Securing Connected Things

Description: This week we discuss Win10 getting a new spring in its step, Microsoft further patching Intel microcode, the U.K.'s NHS planning to update, another hack of modern connected autos, Oracle's botched WebLogic patch, an interesting BSOD-on-demand Windows hack, a PDF credentials theft hack (which Adobe won't fix), your Echo may be listening to you, a powerful hotel keycard hack, a bit of errata and feedback, and a discussion of another Microsoft-driven security initiative.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-661.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-661-lq.mp3

SHOW TEASE: Your hotel room is owned. Your connected car needs help. Your Echo has turned into an eavesdropper. And does Microsoft have the security answer for everything from your light bulb to your nuclear plant? It's all next on Security Now!.

FR. ROBERT BALLECER: This is Security Now! with Steve Gibson, Episode #661, recorded on May 1st, 2018: Securing Connected Things.

It's time for Security Now!. This is the show with Steve Gibson, our very own security guru. He's on a mission of singular focus to make the universe safer and simpler. He's kind of like Thanos but with less purple. I'm Father Robert Ballecer, the Digital Jesuit, in for Leo Laporte, who's currently in Japan learning how to pilot giant robots. Steve, how are you, my friend?

Steve Gibson: I'm great. And fun to be with you for our second of two of your hosting of the podcast while Leo, as you said, is off roaming around Japan somewhere. Last week we talked primarily about an initiative that Microsoft had put together, the Azure Sphere. So I misstated something that we'll be correcting in the errata section of today's podcast, which makes more sense because I was giving them too much credit for something. And so now it's like, okay, now I understand. This does make more sense. But they've been busy.

And there's another Microsoft initiative I want to wrap up today's podcast discussing with you about their sort of conceptual display of how to, like, what's necessary to truly solve, and not just say we did and hope no one challenges us, but what's necessary to truly solve the IoT security problem where it's not just light bulbs, but it's also nuclear power plants and SCADA managed systems. I mean, how do you really, really do it? And they haven't - they're not, like, selling it. You can't buy it off the shelf anywhere. But they produced a very comprehensive sort of executive overview of, like, saying - and they also, in the beginning, say this is not like anything we invented, or rocket science. This is

applying understood cryptographic principles all the way. So I think we'll have a neat discussion about that for this 661st podcast, as we start off May.

But we also need to talk about the fact that Windows 10 just got a new spring in its step - not called the Spring Creators Update, but now officially named and released, apparently a week before it was expected, and causing some trouble. Microsoft also has further patched Intel microcode. This was actually happening while you and I were talking last week. But it's important, so we need to catch up on that. Also we know that the NHS in the U.K., their National Health System, got really hurt by WannaCry. They've decided how to fix that in the future.

We also have another update on the hacking of modern autos with a really interesting look inside and a frightening port scan of a VW. We've got Oracle's botched WebLogic patch which, even if you give a company responsible disclosure, if they then screw up the patch, well, then everybody knows about it. And that's the situation we're in right now with very vulnerable WebLogic servers that are right now being exploited after Oracle said, okay, we got this fixed. We also have an interesting Blue Screen of Death-on-demand Windows hack, which even Microsoft said, eh, okay. A new PDF credential leak, which Adobe has said they won't fix. Also a little bit of a problem with the Amazon device - I'm trying to remember, can I say "Echo"?

FR. ROBERT: You can, you can.

Steve: I can't say the "A" word. But the Echo is safe. It may be listening to you. Which of course has everyone freaked out all the time. I can't remember if it's Lisa, Leo's wife and CFO of TWiT, or if it's other guests. I think there are some anti-IoT listening device people within the TWiT network. Anyway, they're like, no, no, no, I'm not having any of those speaker things because they might be listening. Well, it turns out yes.

Also, tying in to the Picture of the Week, you and I are going to have fun talking about a very powerful hotel key card hack which, after a short bit of time, produces a master key for that entire facility, allowing any door to be opened. We have a bit of errata, as I mentioned, and some feedback. And then we're going to talk about this very sweeping proposal that Microsoft has put forward about how to actually make sure bad guys don't go in and turn off the coolant on the reactor, which you really do want to avoid having happen.

FR. ROBERT: Cortana, please open the pod bay doors. Now, Steve, this is a packed, packed episode. And not only is it a packed episode, but you've got some stories in here that I absolutely want to get your feedback on.

Steve: We're going to have fun.

FR. ROBERT: Especially the key card hack and some of the Echo on-demand stuff. So let's get to it. All right. I skipped the Picture of the Week last week, so I'm not going to do it again this time.

Steve: You're a quick study, Padre.

FR. ROBERT: Especially since it's cool. It's hardware. I'm a hardware guy. Steve, what is this?

Steve: So it ties into the story we'll be talking about, about the hotel key card hacking, which the F-Secure guys apparently spent a decade on. Now, it's not clear that they

actually did nothing else for 10 years because that's a long time. But the story is that one of them 10 years ago had a laptop stolen from a hotel room. And that's, of course, 10 years ago laptops were more expensive and more cherished and prized. And it was a security guy's laptop, which might have had crown jewels and stuff on it. So that got them thinking about the security of hotel rooms and the movement from physical keys to the arguably more convenient for everybody, if they were equally secure, new key card technology that you stick into the slot or you swipe or whatever you do.

So here we are now, 10 years downstream. And as we will be talking about when we get to it a little bit later in the podcast, they completely cracked the system so that, after sniffing a facility's card, they're able to synthesize a master key for that facility that opens every door there. So, interesting. This is just a picture, a cool picture of the hardware that they used, showing an RFID-style power and signal loop antenna hooked to some cool little - these things are all over the place now - small little battery-powered processor I/O device.

FR. ROBERT: Yeah, it looks like an embedded processor with some sort of software-defined radio shield on it.

Steve: Yeah. And it's not clear, I guess, I mean, you need to power the RFID, so you've got to have enough juice to get it going. But you're right. We can see an antenna connection, and it looks like it loops around the back and then comes back forward where it plugs into the loop antenna. But anyway, it's got a couple little blue LEDs because you've got to have those. And it looks neat. So we'll be talking about the details of the hack that they pulled off. But I thought that would make sort of a perfect little Picture of the Week for us.

FR. ROBERT: And it's a tease. We like teases.

Steve: And we do.

FR. ROBERT: All right. Now, you did mention that you gave Microsoft too much credit last week, so let's go ahead and take some of that back.

Steve: Yeah. Okay. So first of all - okay. We've been talking about the continually unnamed, upcoming, next feature update to Windows. We have, as we know, there are periodic security updates. And then there are the bigger, what, are they supposed to be semi-annual, I think twice a year officially, feature updates.

FR. ROBERT: Right.

Steve: The previous one was the Fall Creators Update for 2017. And we didn't know if this was going to be the Spring Creators Update. It turns out they're just calling it the April 2018 update. And most people were expecting it to land on the 8th, which would be the second Tuesday of the month, this first Tuesday of the month being May Day, May 1st. So essentially this is all happening as early in the month as it could. But people got surprised. It actually hit yesterday, making it on the last day of April rather than the first day of May. So this is Windows 10 v1803. Until now we were at 1709. So we're now at 1803 with the most recent feature update. And there's, like, a whole bunch of crap, oh, I mean, I guess I could say that.

FR. ROBERT: Stuff.

Steve: Stuff.

FR. ROBERT: Yeah, stuff.

Steve: New features. New features in the April 2018 update's timeline, and it will walk your dog for you and things you may or may not have an interest in. There's a lot of criticism of it. There are problems already that people are having. Lots of buzz about how not to have it stuck in your machine. Apparently, well, and I did it on purpose because I have to, to see what's going to happen to me. And it took maybe an hour and a half. And afterwards I used the new storage whatever it is feature to clean up because there was always the disk cleanup that we've talked about. And Microsoft decided to give us, like to take more responsibility for that. And so I ran that afterwards, after making sure that it didn't seem to have broken anything, and I got 8GB of hard drive back on a system that was previously already cleaned. So once you're sure that you want to stay with this version 1803, there's 8GB waiting for you to recoup, if you're interested.

The thing that, I mean, I'm not a big Win10 fan. I have Win10 machines because I need to make sure that the little software gizmos that I produce work there. And in fact it turns out that in the case, for example, of InSpectre, they screwed up the, what was it, the 64-bit version of one of their updates so that you couldn't, from a 32-bit code, you got different results as a 32-bit app compared to a 64-bit app in querying the new API. And so I was forced to write my first-ever 64-bit MASM as a little stub which is built into InSpectre so that it runs that to do its probing, if that's where it finds itself. So, yeah, I have Win10 systems, although they're not my main world yet.

Two interesting things. Strangely enough, and we've already - we've been seeing Microsoft hating Linux less recently. They've made some interesting moves in that direction. This version, this 1803, has built-in the TAR and CURL commands.

FR. ROBERT: Yup. Yup.

Steve: And so that's like, oh, that's nice. The thing that I liked, because it's annoying to me when I get these little slide-out, what do they call them, toasts or something, the new messaging API that kind of stuff slides out down the lower right?

FR. ROBERT: Yeah, that's one of the first things I kill.

Steve: Yeah, new version of Candy Crush Saga. It's like, okay, what?

FR. ROBERT: I can't stand them, the little notification applets, yeah.

Steve: So there's something called "Focus Assist" that you can get to through the Control Panel under System. Focus Assist basically allows you to keep your focus rather than having it assist you.

FR. ROBERT: And we're talking about human focus, by the way, which is, I mean, seriously, there's nothing that will break your concentration, like you're in the zone, programming, writing, whatever it may be, and then you get a pop-up for mail, or there's a new tweet. I mean, it just breaks you out of that. It's nice that finally one of the manufacturers said, hey, you know what, what if we gave them an option to remove the distractions?

Steve: Yes. So you can, without any fancy stuff, you're able to say only show me dire warnings where my computer or I am near death. And what I do is, because you can do a time of day where you want to keep your focus, I say, okay, turn that on at 2:01 a.m. Turn it off at 2:00 a.m. So I have a one-minute vulnerability where my focus may be

disturbed, but otherwise it's on all the time. Anyway, there's lots more. I would say you probably know, if you have Windows 10 Pro and Enterprise and one of the other 12 other than Home, you are able to delay feature updates for up to 365 days. I know that because I once dialed it all the way up to maximum, and that's where it stopped was at 365. And there are other things you can do if you have a version of Win10 Home, and you really don't want this. The point is, if you do a check for updates, and presumably on Tuesday of next week, on May 8th, I mean, this has moved into the update channel rather quickly.

FR. ROBERT: Yeah, they just push it.

Steve: And it's, yes, they're expected to push it. And the biggest takeaway is people are having problems. There are all kinds of problems after 24 hours. So make sure this is what you want. You do have 10 days, I think it's 10 days, to back out of it. As long as you don't make lots of other changes and things, you can back out safely, if it ends up installing itself on you, and you regret that.

FR. ROBERT: Steve, I updated this laptop here to the Pro, the Enterprise version of Windows, specifically so I could tell the automatic updates to go stuff themselves. And it's amazing. They understand that it's annoying, so they give the higher end customers the ability to turn that off. But everyone else just gets it downloaded automatically. Yay. Yay.

Steve: Okay. More on updates. Also, as I mentioned while you and I were talking last week, Microsoft released a round of updates to their microcode update packages. We have talked extensively about KB4090007. That was a memorable number. That's 4090007 which was the Windows manual install microcode update only for the latest version of Windows 10, which at the time until now was 1709. This was first released on the 13th of March, and then the next day related releases for 1703 and 1607, the successively earlier builds, were released. But there was no release for 1507, which was the still earlier one. So as of last week, for our listeners who don't already know, and you're not being updated or notified or anything so there's no reason you would. And following what we discussed a couple weeks ago, which was Intel's final statement that, okay, here's all of the processors by CPUID that we are going to and have patched for the Spectre v2 vulnerability. Oh, and a couple that we've decided we're not going to.

So as we know, when that happened, I incorporated that list into an updated version of InSpectre so that it would be able to tell people whether there was a microcode update available somewhere, whether it was going to be from your BIOS supplier through your OEM that you bought your machine from or, hopefully, from Microsoft. So as of last week, that is, 4/24, we have updates for the 1507, 1607, 1703, and 1709 - much more comprehensive than what we got in March, but still not complete. So I have a little processor, it's a 406c4, and it's on Intel's list of "has been patched," but it is not yet in Microsoft's patch update. So presumably they will get this done, that is, they've already demonstrated a willingness to rev these KB articles.

I've got all the Knowledge Base numbers in the show notes for anyone who's interested. And 4090007 is still current, I mean is still the proper Knowledge Base article, if you are at the previous Fall Creators Update version of Windows 10, 1709. So presuming that most of our listeners are there, you can still just google "KB4090007" and then scroll down to the bottom to get the link to the catalog, the update catalog. And then you'll see four entries. You'll see the older ones that were released on March 13th for x86 and x64, and then the newer ones released last week on the 24th, also for x86 and x64. So hopefully we will eventually get full coverage from Microsoft for all the processors that Intel has done the microcode update engineering for. Certainly it wasn't easy for them to

do that. And in fact it turns out they probably couldn't for some.

Oh, and there is something notably missing. I forgot to mention that, even though there are microcode updates for 1507, there isn't one for the 1511 version because then it jumps to 1607. And I'm sure Paul and Mary Jo, who follow this more closely than I do, would know what 1511 is and why Microsoft just said, eh, no, sorry. But at some point Microsoft's going to make you go up to something newer anyway, where you will be able to use the updates.

FR. ROBERT: My guess is that it's one of those niche versions for some sort of embedded device. Steve, actually I had a question about this. Now that these are starting to come out in earnest, and we think we have a version of the patches that's actually going to work, that will actually be usable, is there going to be a cat-and-mouse game? I know they're updating the microcode. But as soon as these hit, people are going to be disassembling the binaries to figure out exactly what they patched and what they did, and they're going to be looking for ways around this. Is this something that's going to have to be repatched and repatched and repatched? Or will the firmware fix, is it disabling something that it's just going to not allow exploits in the future?

Steve: So what it's giving us is it's giving software more control over some features which were believed to be safe. So we discovered, unfortunately, that it was possible due to all of the optimizations in the processor to deliberately bias the branch prediction logic in one direction, and then use that to perform a test that could create some information leakage.

FR. ROBERT: Right, a response-type test.

Steve: Yeah. So the worst that, I mean, and it's worth noting also this was all, I mean, an awful lot of fur flying for theoretical problems. And I salute the industry for taking this seriously because we know, had the news gotten out and it not been taken seriously, there would have been exploits developed. But as far as we know, there haven't been any. I mean, as far as we know, not a single other than theoretical proof-of-concept, look, we got a few bits - although I do remember seeing a rather surprising, I guess you'd call it "bandwidth" of information leakage. That is, the number of bits that you could suck out across a process boundary was surprisingly high. It was like, whoa, okay, you could actually do some damage at that rate of exfiltration of data. And you were able to determine where you pulled the data from. So if you knew of which areas were sensitive - and you could also use it presumably to probe the ASLR in order to derandomize the Address Space Layout Randomization by determining what was where in memory.

And so, I mean, it's becoming increasingly difficult to actually pull off these things because over the last decade security really has gotten a lot better. But in the worst case, this was an information leakage across process boundaries. The point is you could turn off your own branch prediction mitigation in your process, but there's still no way that we know of to turn it off in the process that you're trying to get information from.

So I think that terminates any cat-and-mouseness. I mean, maybe someone will come up with a way of not getting the microcode applied. I mean, there's that possibility, if you did some sort of like you got a root in, like a rootkit-ish attack, because after all Microsoft is - it's a Microsoft driver loading early which is patching the microcode on the fly. So if you were able to get in somehow in EFI or in the BIOS or in the boot sequence early and subvert that, then you could keep the microcode patch from being applied and then go about your business.

FR. ROBERT: Is this something that you can actually make UEFI aware, just so that the system knows if someone's trying to disable the launch of the updated firmware at launch?

Steve: Well, yes. Secure Boot operates from a trust anchor which cannot be compromised. And we'll be talking about trust anchors at the end of this podcast because that's the core of Microsoft's idea for how to keep somebody from turning off the coolant on your nuclear reactor. So as long as you've got Secure Boot, and as long as it is secure, that is, I mean, as long as the guarantee that it offers is met, then we're covered. And remember that the BIOS is patching the chip also. That is, already we've got BIOSes patching our processors. Microsoft is just doing a downstream repatch of what the BIOS has probably already patched because the BIOS doesn't know about the newest stuff and probably never is going to.

FR. ROBERT: Yeah, it's a whole lot easier to download a patch than trying to instruct your users on flashing the BIOS so that it loads the right microcode.

Steve: Yes, yes. And we know that users can get in trouble, too, if the dog trips over the cord while the EPROM is being flashed. It's like, oh, now my entire system is dead. It's down hard.

FR. ROBERT: There are some people who are going to look at this, and they're going to say, oh, well, all of that was just panic. It was overblown. I don't see it that way. I'm with you. I think if this had not received the attention that it received, Intel would have just quietly kind of brushed it off, and they would have said exactly what we're saying right now, which is the possibility of using this exploit is so low we don't think it's - you don't really need to do anything. But once someone knows that it's there...

Steve: There is a problem, well known, with proving a negative. I'm put in mind of the W2K scare, the whole - I mean, I'm sorry, Y2K, the Y2K. Everyone ran around, like, oh my god, oh my god, oh my god. When we go from 1999 to 2000, the entire computing infrastructure is going to collapse. Well, it's absolutely for sure that it would have, had we not spent the previous six months leading up to that actually fixing all of the problems. And so when nothing happened, I mean, I was deliberately awake with my friends on that particular New Year's Eve. I haven't bothered since. But it was like, okay, what's going to happen? And, like, eh, nothing.

So there were all the naysayers who said, ah, see? This was all a big scare. No, you know, those of us in the industry know that there were actually a huge number of problems widely distributed that would have really caused havoc if there hadn't been some attention paid to it. So this is like that, I think. People say, oh, it really was never anything. It's like, well, no, it probably could have been something.

FR. ROBERT: That's the old, wow, you spent all that money on securing your house, and no one ever broke in, so obviously it was a waste of money.

Steve: Exactly. All those automatically turning on floodlights, you just didn't need those, even though they did turn on a few times.

FR. ROBERT: A couple, yeah. Now, Steve, last year I think quite possibly one of the biggest stories, I mean, yes, you had this Intel kerfuffle. But one of these stories across that magical boundary between the technorati, the digerati, the geeks, and the normals was WannaCry because it brought out the worst fears that someone is going to lock your files, and you won't be able to get access to them. That was one of the highest viewed

videos here at TWiT. We did a thing on Know How.

Steve: And it made you - and it made you - WannaCry.

FR. ROBERT: WannaCry. A lot. A lot. And then there was a lot of political intrigue. But one of the stories that really started the political and press move was the Department of Health and Human Services or whatever they call it over in the U.K. having a large number of their workstations locked out by WannaCry. And you say that they've figured out a way to not have that happen again.

Steve: Well, I just got a kick out of this. This is a quickie. We don't need to spend much time on this. But I did note that the U.K. health agency, the official name is the U.K. Department of Health and Social Care, has announced that it will transform all NHS [what they call their National Health Service] computer systems to - wait for it - Windows 10.

FR. ROBERT: Tada.

Steve: Officials cited the operating system's more advanced security features as the primary reason for upgrading current systems such as Windows 10 SmartScreen technology, which we know is built into Microsoft Edge, and Windows Defender. And the coverage for this was in Bleeping Computer. And of course they're our ransomware guys. They have been literally on the bleeding edge of ransomware. That's where they first came to light, really, although I guess they'd been around for about 10 years. But it was interesting that Bleeping Computer characterized Windows Defender as "Microsoft's sneakily good antivirus product."

FR. ROBERT: Sneakily?

Steve: Sneakily. So I thought, oh, I mean, this is coming from a source whose opinion I trust on something like AV. And the writers at Bleeping Computer, this was their terminology. They characterized Windows Defender as "sneakily good." So just keep that in mind, you people who are still installing third-party junk.

FR. ROBERT: You know, us hardcore Windows users have been using Defender for years now, and most of us have not installed an AV because Defender actually works a whole lot better than those third parties. Just saying. Just going to throw that out there.

Steve: Yes. I'm with you. Thank you for making it more clear for those listeners who are saying, wait, what is he trying to say? So anyway, that's really all I had to say here was that they were - we know that most of the WannaCry-vulnerable machines were Windows 7, but that's because there are so many of them. What NHS still had was XP, which really upset people. But again, their medical systems, they had the, well, if it's not broke, then we'll not fix it next month. Just like let's leave it alone. And of course they got hugely hit. Basically the entire U.K. health system was offline as a consequence of it having swept through them. So they're moving to Windows 10. And I guess they missed the upgrade Window, didn't they. So I guess they're going to have to pay for it? Or Microsoft will [crosstalk].

FR. ROBERT: You have to remember that the U.K., they actually cut a huge deal, multimillion-dollar deal with Microsoft to continue support for XP.

Steve: Right.

FR. ROBERT: So, I mean, they're back in the news. It's sort of like, maybe updating to a

new operating system isn't a bad idea.

Steve: Maybe it's time.

FR. ROBERT: Maybe, maybe a good time.

Steve: It's only been, what, 10 years.

FR. ROBERT: And, okay. I give Microsoft a lot of guff because I have found my machines updating at the wrong time pretty much on a regular basis. And it's normally right before I start Know How, when it's the laptop I only use once a week.

Steve: And you get the little, I call it the "stretchy rollercoaster."

FR. ROBERT: Right, exactly. And it's one of those situations where, even though I have active hours set, it's gone so long between updates that the operating system just says, no, I'm not even going to give you the option anymore. We're doing this now. But there is some merit to that. The fact that my computer can be used in a nefarious way against another network, it kind of says, okay, well, if you want this to be connected to the Internet you...

Steve: With that comes some responsibility.

FR. ROBERT: Yeah. It's like immunity, group immunity here. It only takes one person who says this is all a crock to hurt the herd.

Steve: Yeah.

FR. ROBERT: Speaking of hurting the herd, Steve, two years ago was it, at Defcon, Black Hat/Defcon, there was this wonderful presentation by two guys who were not security people, they were just programmers who went on to be security people, showed not just a proof of concept, but a demonstration of taking over a vehicle through its Internet connectivity. And the world has said, okay, well, now we've learned, that's never going to happen again. Unfortunately, it has; right?

Steve: And this is a meaty story, so let's make this a tease. Tell our listeners about our second sponsor, and then we're going to get into some really nice research that was recently done and found some vulnerabilities that are, well, that's made their manufacturers scramble.

FR. ROBERT: Scrambled manufacturers. I think that's enough of a tease to me. All right, Steve.

Steve: Okay, Padre. Before we start this, I just have to say I was doing a little bit of poking around just now, wondering what the problems were with yesterday's update to Windows 10. Martin Brinkman over at Ghacks has some nice coverage of this. And get a load of this.

FR. ROBERT: Uh-oh.

Steve: There's some enumeration here. First of all, we were talking about Focus Assist. So one of the problems people are having is slow alt-tab performance when tabbing out of games. The fix? Disable Focus Assist in the options under System Focus Assist. But the bigger one, Martin writes, no microcode update to protect the system against Spectre

attacks. Can you believe this? He says Microsoft released updates only for Windows 10 v1709 and earlier, and has not incorporated the updates into Windows 10 v1803. If you installed the update on the PC in an earlier version and ran the update afterward, you will notice that the PC is no longer protected.

FR. ROBERT: You know, it sounds like someone could have taken a USB drive from one side of Microsoft to the other and just said, hey, can you include this in the patch? Yeah, thanks.

Steve: Oh, my god. And this list, I'm just scrolling down through this nightmare list. It's like, you've got to be kidding me. Okay.

FR. ROBERT: I mean, they don't do these very often. You figure that they'd actually sit down with everyone and say, is this ready to go out? I mean, that was the whole promise of this continual patching process, which is, look, we'll get it right.

Steve: Wow.

FR. ROBERT: Okay. Now, you know what, Microsoft has done some good things. I'm going to just tell them that's a bonehead thing, and we'll move on.

Steve: Yeah. I really did like Windows 3.1. I thought they nailed it on that release of Windows.

FR. ROBERT: I still have a copy of both Windows 1.0...

Steve: Good for you, yup.

FR. ROBERT: ...and IBM OS/2.

Steve: Nice.

FR. ROBERT: And I think I still can't install OS/2 because it would run too slowly.

Steve: Okay. So back to business. Anyway, people may, if you're interested in maintaining your security protection, you may want to wait until - I guess it'll be a new KB because they have a KB per version. So hopefully there will be a Knowledge Base manually installable update for the Spectre v2 patches for v1803 of Windows 10. My goodness, Microsoft.

FR. ROBERT: Right.

Steve: Anyway. Okay. So it's called the CAN bus, C-A-N, which is a convenient acronym since, once you gain access, you CAN do anything you want. The CAN bus appeared and became standardized during the '90s; and it's now present, it's become like the way things interconnect in our vehicles. It's present in every automobile built since because every component in today's cars are hooked up to this single bus. It's used to control everything in the vehicle from steering to unlocking doors to the volume of the radio, the fuel/air ratio mixture, I mean, the transmission, everything.

FR. ROBERT: ACU, power steering, power brakes, basically everything in the car is connected to the Control Area Network.

Steve: Yes. It's a straightforward protocol, which is part of the reason why it succeeded.

FR. ROBERT: Not encrypted.

Steve: Correct. Each message has an arbitration ID and a payload. No authentication, no authorization, no signing, no encryption. It's like rev point one.

FR. ROBERT: What could possible go wrong with that?

Steve: Exactly.

FR. ROBERT: See, Steve, here's the thing. Here's the thing, though. That was fine. The CAN bus was fine because cars were isolated.

Steve: Yes. Yes.

FR. ROBERT: But then they were connected. And auto manufacturers never thought, hey, you know what, maybe that connected device shouldn't be connected to the completely unauthenticated, unencrypted network inside the car that runs everything.

Steve: And to their credit, I mean, after the first generation of problems, they said, ooh, maybe we shouldn't have one bus. Maybe we should have a couple buses. And in fact there are some use cases for a high-speed bus where you need much closer to real-time stuff like, oh, I don't know, braking, steering...

FR. ROBERT: Airbags.

Steve: ...collision avoidance radar, that would be handy to have a low latency on. And then things like door locks and seat movement back and forth. I mean, everything is on the bus. So there's slow speed and high speed. What you would wish was that they were isolated. But there's a gateway that interlinks the buses. So we don't have complete isolation because that was just a bridge too far. So, yes, it's wide open. Once you're on the bus you can send arbitrary messages as an equal peer. It's just a complete peer communication backbone. Send arbitrary messages which will be received by all parties connected to the same bus. There's no sender or recipient information, and each component decides for itself if a message applies to it. So exactly as you said, and as we often say on this podcast, what could possibly go wrong?

So a couple researchers working for a group, Computest, in the Netherlands, decided to sit down and see what they could do. In their research they wrote: "We started this research with nine different models from nine different brands. These were all leased cars belonging to employees of Computest. Since we are not the actual owner of the car, we asked permission" - which I thought was very nice - "for conducting this research beforehand from both our lease company and the employee driving the car." Hopefully not while they were driving the car.

"We conducted a preliminary research in which we mapped the possible attack vectors of each car. Determining the attack vectors was done by looking at the architecture, reading public documentation, and by a short technical review." They said: "Things we were specifically searching for were cars with only a single or few layers between the cellular connection and the high-speed CAN bus; cars which allowed us to easily swap SIM cards; and cars that offered a lot of services over cellular or WiFi." Meaning if it was a recent car, but didn't have much in the way of radio link feature set, one would expect it to have less opportunities, basically a smaller attack surface.

And then they said: "From here we chose the car which we thought would give us the highest chance of success." And they go on to describe their attack. I wanted to show you, though, Padre, in the show notes, on the next page from where I am is the result of an Nmap port scan.

FR. ROBERT: I love this. Actually, by the way, this is how the original duo did it. So Charlie Miller and Chris Valasek, they started doing port scans, and that's how they figured out, oh, my gosh, everything's open.

Steve: Yeah. They found port 23, telnet is open, is like responding to connections. I mean, and there's a whole bunch of - 23 is the only one down in the service port range. The other ones are like 10123, that's open. 15001, that's accepting connections. 21002, 21200, they look like they were kind of chosen by programmers who said, oh, let's just choose, you know, 22111, 22222.

FR. ROBERT: Aside from 23, none of these are standard ports, and no one will be able to figure out what they do.

Steve: Yeah. Oh, no one will even think of looking up there. And then 49152.

FR. ROBERT: My favorite.

Steve: Yes, that had Universal Plug and Play.

FR. ROBERT: What are they doing?

Steve: Yes. You want UPnP in your car. Oh.

FR. ROBERT: Oh, and by the way, the default credentials for that are "admin" and "linksys." So, you know, just FYI. That's horrible.

Steve: I know. They said: "After further research, we found a service on the Golf" - they ended up choosing a Golf from among their nine starter cars - "with an exploitable vulnerability. Initially we could use this vulnerability to read arbitrary files from disk, but quickly could expand our possibilities," they write, "into full remote code execution." Yes, run code on the car's processor. They said: "This attack only worked via the WiFi hotspot, so the impact was limited." They said: "You have to be near the car, and it must connect with the WiFi network of the attacker." But, they said, "But we did have initial success." And then they show obtaining a shell, a dot slash exploit. And the IP was 192.168.88.253. And they show it says, "System seems vulnerable. Enjoy your shell." And then they do a uname -a. And sure enough, QNX is the little version of OS running on this machine.

FR. ROBERT: Now, Steve, a little background for our audience. The reason why this only works through the hotspot is not because of any engineering that they did. Back to the Miller and Valasek hack, after they showed the world what was possible, the auto manufacturers worked with the carriers to make it so that you couldn't just randomly ping those devices. But we have, like Emily the Strange in the chatroom is saying that's why she has all that stuff turned off on her new car. It doesn't matter. The device is still active. It's still there because they need a way to be able to turn on the service if you happen to start paying for it. So that device is still in your car, and it is still accessible, but this hack was done locally. So they'd have to actually be within WiFi range. If it wasn't for those carriers, however, and if you happened to change to a carrier that

doesn't block that port scanning, you are now vulnerable over the Internet again.

Steve: Yeah. So they did not disclose - they were responsible. They did not disclose publicly what they found. But Volkswagen is scrambling. They said in their conclusions: "Internet-connected cars are rapidly becoming the norm. As with many other developments, it's a good idea to sometimes take a step back and evaluate the risks of the path we've taken and whether course adjustments are needed. That's why we decided to pay attention to the risks related to Internet-connected cars. We set out to find a remotely exploitable vulnerability which required no user interaction in a modern-day vehicle, and from there influence either driving behavior or a safety feature."

They said: "With our research, we have shown that at least the first is possible. We can remotely compromise" - and then they used some acronyms, the MIB IVI, which is sort of the vehicular entertainment and extra features side - "and from there send arbitrary CAN messages on the IVI CAN bus." Which, again, is not the sensitive one. It's the more entertainment side. But it's also the one that shows your dashboard instrumentation. And as we know from the previous hacks, you can do things like put up "Hi there, how you doing, fella?" messages where it's supposed to be showing you your remaining miles of gas and so forth.

And they say: "As a result, we can control the central screen, speakers, and microphone." They said: "This is a level of access that no attacker should be able to achieve. However, it does not directly affect driving behavior or pose any safety risk due to the CAN gateway," which we mentioned is isolating multiple buses within the car. "The gateway is specifically designed to firewall CAN messages, and the bus the IVI is connected to is separated from all other components. Further research on the security of the gateway was consciously not pursued." And actually they did some moralizing later on about proper conduct and what you need to do and advice for people who are doing this kind of research so you don't end up being jailed and so forth.

But anyway, we are driving computers, and they are becoming connected computers. And in a way they're a little IoT-like, inasmuch as the manufacturers just want to sell cars. They don't want to sell computers. People want the features of mobile computers, so that's what they're selling. But they're certainly paying attention to security. And the good news is this kind of coverage keeps the automotive manufacturers focused on that need, which is all for the best.

FR. ROBERT: You know, Steve, ultimately this is just a pivot attack. This is the ability to get into a device, own the device, because now you can run code on its processor, and then you can connect to the other side. And the other side is the CAN bus. We had talked about this at the last Defcon, and one of the ideas that I kept pushing was I said, look, the reason why manufacturers want this is because they want access to the error codes on the CAN bus because that's where they can give you your value-add. And so that's why they're so reluctant to separate that entertainment system from the CAN bus, because if they do that then they don't have that value-add.

I said it actually would be trivial for them to create an interface that only allows one-direction communication. And it really should only be one direction. There's no reason why the entertainment system should be issuing CAN bus commands to the rest of the vehicle. So you can set it up so that the CAN bus can issue error codes to the entertainment system, but nothing can come back. That's just a simple firewall.

Steve: Right, right.

FR. ROBERT: And the fact that they haven't done something like that yet, it really kind of

tells me that maybe they're not putting as much into security as they say they are.

Steve: Yeah. Well, and again, features versus security. They want to have the features. I had a car - I don't remember now if it's the one I'm still driving, it might be - that was firing off some spurious "service engine soon" warnings, like the little idiot light on the console would just come on. And I remember taking it in once, and there was nothing wrong, and so they reset it. And then a few weeks later it turned on again. I thought, okay, this is dumb. So I just got myself a dongle and reset it myself. So that was kind of cool that I was able to do that.

So it's the power of that bus that gives you features. But you sure do wonder where they are from a security standpoint. And we also know the famous example was trying to have encrypted keys on a DVD. If you're going to play the DVD in the living room, it's got to decrypt it. So the keys are in the DVD player. Similarly, there's nothing you can do, like having a secret password or secret keys in a car that only the service technician's equipment has because, if the keys are in your car, hackers can get them. So that's futile.

FR. ROBERT: Isn't it kind of strange, Steve, that since the early days of what we're doing right now, the topics haven't really changed? The technology has gotten more interesting, but it's still this whole idea of does it matter what kind of security precautions you put into place? If you leave the key in a public area, someone's going to own it. There's no way around that.

Steve: Yup.

FR. ROBERT: All right. Let's get off of VWs because they've had enough hardship over the last couple of years.

Steve: So it's interesting, a hacker does the right thing with a major vulnerability, a horrible remote code execution vulnerability, discovers it, notifies responsibly the party who's in charge of the software, the owner of the intellectual property, in this case Oracle, and says, oh, you've got a problem. This was last November. After a few months a patch is released to fix this problem. Then it turns out that it was misfixed such that it's possible to work around it.

So that's where we are at this moment in time. We discussed this problem maybe a month ago. There was a highly critical flaw in Oracle's WebLogic servers which was responsibly disclosed. We learned that it was a Java deserialization flaw. And we had never discussed on the podcast serialization and deserialization in Java. Essentially, as we discussed at the time, Java maintains structured objects that is the way Java looks at things as they're highly structured. They have an internal structure. So you just can't, like, send it over the wire.

Serialization is the process of essentially turning this structured object into a byte stream which you can then store or shoot across a network. Deserialization is the other end, taking the serialized descriptor of a complex structured object and sort of rebuilding it, expanding it. Well, a deserializer is necessarily an interpreter. And one of our constant refrains now on the podcast is it is so difficult to get interpreters to be security sound, to have a complex interpreter that doesn't have some exploitable edge cases. And that is exactly what was found in Java's implementation in the code for their WebLogic servers, which allows anybody over the Internet on TCP port 7001 to now remotely execute their own code on Oracle's WebLogic servers. This is now loose and in the wild.

Somebody tweeting as @pyn3rd on Twitter, who claims to be part of the Alibaba security team, has found a way to remotely bypass the security patch which was recently deployed and exploit the WebLogic vulnerability once again. So with any luck, the only thing attackers will care to do is mine some coin. WebLogic server admins should consider themselves lucky that most people these days won't care what's going on within your enterprise or on your server. They just want to make some money.

FR. ROBERT: They just want the processor cycles.

Steve: Exactly. So in that sense there's sort of a - Oracle gets a break because probably all that's going to happen is that the processor will get bogged down, somebody will come along and go, hey, why is this thing generating extra heat or consuming extra power or our processor monitor is showing that something is sucking up cycles. And then they'll reset it and reboot it. Maybe, I imagine, Oracle must be scrambling to fix their fix, and we'll be pushing out another one, this time without the benefit of knowing in advance what the problem is. And as will always be the case, some of these things will never get patched.

FR. ROBERT: Steve, help me to wrap my head around this exploit. So I understand in Java you serialize to create a package. And then that package goes to a deserializer, and that's where you get your code.

Steve: Right.

FR. ROBERT: So is this exploit about messing with the package before it gets to the deserializer?

Steve: Yes.

FR. ROBERT: Okay. So I can take any package, make a couple of changes to it. Now I'm taking advantage of the way the deserializer works to inject my own commands into it.

Steve: Right.

FR. ROBERT: Okay.

Steve: Right. The deserializer was written with the assumption that a valid serialized stream would be coming to it, and it wasn't protecting itself from malicious serialized streams. And being an interpreter, someone cleverly discovered that they could give the deserializer some work to do that would allow them to cause it to execute their code.

FR. ROBERT: Again, this is going back to the DVD conversation and the CAN bus conversation. Some engineer of Java in the early days said, "Why would anyone want to do something bad with the deserializer? They wouldn't be able to do their work."

Steve: Right. It's such a good thing. It's a fabulous feature. It's a benefit.

FR. ROBERT: And these packages are not signed in any way. I mean, Java just assumes that, if it's receives a packet, it's valid.

Steve: Yup, exactly.

FR. ROBERT: All right. Good stuff.

Steve: Exactly. Okay. And what's interesting is that our next story is exactly the same problem in an entirely different context. Somebody on GitHub has hosted an NTFS file system image which will crash Windows. Period. It's a little 10MB NT file system image which he handcrafted to cause Windows to barf and collapse horribly when it attempts to mount that file system. He reported it to Microsoft and said, hey, look at this. I've got an NTFS file system. If you stick it on a thumb drive, and then you plug it into a USB port on somebody's Windows machine, it crashes the machine. And they said, "Yeah."

FR. ROBERT: Wow, you do seem to have that, huh.

Steve: And I'm paraphrasing here. They said, well, yes, having Windows barfing and collapsing horribly is always a bit embarrassing. But the attack, such as it is, as presented, requires physical access to the machine and for connected drives to be automatically enumerated and mounted. Now, for his part, the discoverer of this problem was mostly concerned that even machines in a locked state, even a locked Windows system, it turns out, will enumerate and mount a USB drive that you present it. So you can do this to a machine even that is not interactively online. Now, we know that the first stage in the development of any attack is a crash.

FR. ROBERT: A crash, yeah. Unexpected condition.

Steve: And so once again we have the NTFS file system interpreter which is looking at the file system structure and assuming it's benign and is - I'm always put in mind of Khan saying "We're all one big happy family" when Kirk doesn't raise his shields. It's like, we're all NTFS. Who would want to hurt us?

FR. ROBERT: By the way, he is referencing "Star Trek II: The Wrath of Khan." Don't forget that every starship can be taken over by a simple prefix code.

Steve: That's right.

FR. ROBERT: Oh, I'm sorry, I derailed us. Continue.

Steve: Anyway, so once again an interpreter. He did comment, because I caught myself up with his most recent GitHub postings, that the most recent update he got to Windows did not seem to be doing this any longer. So even though Microsoft said we don't care, maybe they fixed it anyway, which would be nice because it would be better if an NTFS file system image were not able to crash Windows because, again, as we have often said, what follows is a takeover. And if you were able to plug a thumb drive into someone's machine, when they were away and the machine was locked, and remotely run your own code, rather than just having it crash, having it actually execute something that you put in there, that would not be good.

FR. ROBERT: We should mention that there have been USB device exploits for a while, but that it typically is code on the USB drive that you can somehow make run or autorun.

Steve: Right.

FR. ROBERT: This is different. This is different. This is not something on the drive. This is the structure of the file system itself. And as you mentioned, this will run regardless of what state your machine is. It always tries to mount a connected storage device.

Steve: It's, exactly, mounting. It's not auto-executing something on the drive. It's just saying, oh, let's see what you've got. And that's all it takes.

FR. ROBERT: Right, right. Well, let's move from that to another Windows fun thing, NTLM, which actually has been in the news this past year.

Steve: Yes, yes.

FR. ROBERT: Along with the WannaCry scare. It's one of these protocols that probably should go away at some point.

Steve: It keeps on giving.

FR. ROBERT: Keeps on giving. What's going on?

Steve: Okay. So get a load of this. And our listeners will appreciate this. Many, many moons ago, in fact it's probably on those OS/2 disks that you have, Padre, IBM and Microsoft got together and created LAN Manager, or LAN Man, as it's often called. They designed what is by today's standards such a horrifically weak authentication mechanism, the so-called LM hash, the LAN Man hash, that it's hard to believe anyone ever took it seriously.

Okay. So get this. In terms of authenticating requests over LAN Manager, a hash is used. The hash is a hash of the password which is used to authenticate. Passwords are, A, not case sensitive. They're always converted to uppercase before hashing. Password characters are limited to a subset of the ASCII character set. So it's not only that you lose the lowercase alphabetics, you lose a bunch of other stuff, too, that are not valid in passwords. Passwords are limited to - is everybody sitting down? - a maximum of 14 characters. Now it's like, okay, well, 14 characters, not great. But get a load of this. The 14-character password is broken into two seven-character substrings, each hashed independently.

FR. ROBERT: Okay, come on.

Steve: You can't make this up.

FR. ROBERT: Okay. So I've got two seven-character hashes that I know are all going to be uppercase.

Steve: With a subset of ASCII. What could possibly go wrong?

FR. ROBERT: What could possibly go wrong?

Steve: And you really can't make this up. If the password is seven characters or less, meaning that there was nothing in the second set of seven for it to crack into two pieces, the second half's null hash is always AAD3B435B51404EE. I stated that because that's only 16 hex characters, which means it's eight bytes, which means it's 64-bit hash. So you have a 64-bit hash of a maximum of seven characters, which we could cut through like butter in this day and age. And of course we do, which is why this has been deprecated historically, but why it continues to cause a phenomenal amount of trouble because it was once ubiquitous. It was everywhere. And for the sake of backward compatibility, it's still around because you never know when an NTLM LAN Man hash might still be used by some server somewhere for authentication.

So the guys at Check Point - so with that bit of background, the guys at Check Point wrote in their research: "A few days after it was reported that malicious actors could

exploit a vulnerability" - and this was just a few weeks ago - "in Microsoft Outlook using OLE to steal a Windows user's NT LAN Manager hashes, the Check Point research team can also reveal that NTLM LAN Manager hash leaks can also be achieved via PDF files with no user interaction or additional exploitation needed."

According to Check Point researchers: "Rather than exploiting the vulnerability in Microsoft Word files or Outlook's handling of RTF files" - remember, because we talked about this at the time, there was quite a chain of action required with Outlook, then RTF, then an embedded OLE object in the RTF file, blah blah blah. So, I mean, you really had to work at it. Now, no. Much less work necessary. It turns out that PDFs have a feature that allows them to embed remote content. And if the remote content is a document located out on the Internet, the PDF reader will make a request for the document, including the user's LAN Manager authorization hash, in order to have the request approved, thus exfiltrating, exporting essentially, the LAN Man hash to anywhere out on the Internet that the PDF has been configured for.

Check Point believes from their testing that any and all Windows PDF viewers can be induced to exfiltrate the user's NT LAN Man credentials in this fashion. And being responsible, they contacted Adobe, you know, the source of Acrobat and PDF readers. Adobe, always quick off the starting line with security, and of course the home of Adobe Flash, so we know how focused they are on security, responded to...

FR. ROBERT: I'm sorry, I can't hold that in. I was really, really trying to be serious there for a second.

Steve: ...responded to Check Point's private disclosure, saying: "Thank you for checking in on this case. Microsoft issued an optional" - that is, user go get it - "security enhancement late last year that provides customers" - who do decide to go get it, but who does - "with the ability to disable NTLM SSO [Single Sign-On] authentication as a method for public resources." In other words, it's on by default; but, oh, yes, maybe that was a bad idea. So we'll issue an optional security enhancement because we wouldn't want to break anything. Oh, no. Anyway, so they said: "With this mitigation available to customers, we" - Adobe, the security-conscious company - "are not planning to make changes in Acrobat."

FR. ROBERT: Oh, come on. When you know the product is broken, it's time to say, you know what, maybe we don't want to be backwards-compatible anymore. It's time. It's time.

Steve: I know. I know.

FR. ROBERT: But on the upside, what I've learned today, Steve, is that, for example, if I have a 16-bit password, two eight-bit passwords are just as good. Right? I mean, if my math is right here. This is a learning show, Steve. We just learned something.

Steve: Yes. We should make it clear to people - actually, I'm put in mind of the WPS flaw. The WPS flaw is the push the button on your router in order to automatically pair your new device with your router's WiFi without needing to worry about a password. It was an eight-digit code. Ooh, secure, okay? No. Because the protocol was broken so that you could separately test the first four digits and see if you got those right, and then test the second four. Oh, except that the last digit of the second four was a check digit, so you always knew what that should be. So you only really had to worry about the second three of the second half, or the first three digits of the second half. Anyway, the point you were making clear, Padre, is that, if you have this 14-digit LAN Man password, you're able to hash the front seven separately from the back seven, assuming that it was 14

characters, and the hash was only 64 bits anyway. So not difficult.

FR. ROBERT: Yeah. For the math people out there, if you have a 16-bit value, that's 65,536 possible values, versus two eight-bit, which are 256 each, so you'd have 512 total values. So, yeah.

Steve: And you also have that extra kick of encouragement when you crack the first half. It's like, ooh.

FR. ROBERT: I got it.

Steve: I'm halfway there. Whereas, if you were having to do the whole thing it's like, well, this could take forever, and I'm not getting any encouragement.

FR. ROBERT: Can you imagine if you had a keypad to enter your house? And the keypad, if you got the first two digits right, it said, hey, you got the first two, only two more?

Steve: Getting closer, getting warmer. Stick with those first two, and now you only have to worry about the next two.

FR. ROBERT: Exactly. Okay. The funny thing is I've actually known some of the people who have developed stuff like this, and they normally have a reason why they did it. Constraint on resources or backward - actually, the biggest one is backwards compatibility. But you kind of want to slap them and say, "Don't do that again. Bad, bad. Stop."

Steve: Yeah.

FR. ROBERT: All right. Before we go on to the good news - because we have to talk about how our most cherished devices might be spying on us. Can we do that next?

Steve: Yes, let's do that next.

FR. ROBERT: Let's do that next. We'll get back to our future lords of IoT. Now, Steve, the Echo was a breakout product not too long ago. It was the center of Amazon's new voice assistant market. And even at that time there was a worry that the devices are listening to us. And now we actually know, well, of course they are.

Steve: Yeah. So a group named Checkmarx has successfully demonstrated a persistent listening skill for the Echo. As we know, the Echo is extensible. Amazon produces and publishes an API which developers can use in order to create new skills which users can download and add to their Echo in order to do things. What they did was created a proof-of-concept which to me seems sort of like, wait a minute, you mean it's vulnerable to that?

They created a calculator app which appears to be a voice-driven calculator. Okay, I guess that's a skill. But when it launches, it spins off a secondary background non-terminating task which causes any such skill-enhanced Amazon voice assist device, because there are a bunch of different ones, you know, different sizes and shapes now, to indefinitely record all surrounding audio, eavesdropping on users' conversations while also streaming the audio to a third-party server somewhere.

So, I mean, it's like everyone's worst fear about what could happen if you have an Internet-connected microphone, like deliberately in your midst. Presumably located

somewhere, like it's not in the closet, it's out where it can hear everybody because that's what it's supposed to be doing. They reported the issue to Amazon, and the company has addressed the problem, though not very robustly, in my opinion. In the reporting of this, they said they now regularly scan for malicious skills that, quote, "silently prompts or listens for unusual lengths of time," then kicks them out of their official store.

And I don't know. To me, for something as important as the security, this feels like an immature and not very well thought out AP. No skills, that is, these add-ons, should have any such access, direct access to the system's microphone. Rather, any skill should register its microphone usage and access policy, whatever that is, and its needs as part of an unhackable signed policy packet, which Amazon approves, then the developers incorporate into their skill, and then it can't be hacked afterwards. And then the device's API should grant the skill's requests, which are filtered through and enforced by the previously declared and approved policy.

I mean, this is how you solve that problem. These are not hard things to do. And I'm just sort of - I'm surprised that in this day and age an obviously security-conscious company could produce an API that allows a skill to establish a background task which leaves the microphone running and streaming data to the Internet. It's like, how do you do that?

FR. ROBERT: Right, right. Yeah, I mean, my understanding was the voice activation part was tied directly to the operating system, and it did submit information to Amazon. But you're right, why would a skill ever need that? Because a skill is triggered by a voice prompt. It's not as if it's running, waiting for the voice prompt.

Steve: Right. Right. It's just like, okay, let's hope they get this right. And probably, I mean, what they would have to do now would be to reengineer the API with additional security, maybe taking the lessons that they've learned, because I'm sure they've learned some lessons since its release, and just do a v2 API and give developers some time to migrate their skill to the v2. And then just shut down access to the v1 API that is obviously not secure. I mean, the idea of, ooh, we didn't realize that skills could listen persistently and stream audio over the Internet. So we're going to periodically scan to see if that's happening. What? No.

FR. ROBERT: We can't keep people out of your home, but we'll drive by every once in a while to see that the lights are on.

Steve: That's right. That's perfect. See if everyone's still breathing happily. My lord.

FR. ROBERT: Now, I got an Echo early on. And I hook it up every once in a while because there are some projects that I love doing with the Echo. However, one of the very first things that I did when I got one was I set up an AP running through a Tap, connected the Echo to that AP. And what concerned me was all the outbound traffic that I saw when I wasn't doing anything.

Steve: How noisy it is, yes.

FR. ROBERT: And that really, really concerned me. Now, it's gotten better, so obviously they've improved their software. But I'm betting, if I were to slap the Echo back on this Tap and run one of those skills, I'd be very enthusiastic about the data that I'd be receiving.

Steve: Okay. So hacking, well, I don't want to call them smartcards.

FR. ROBERT: Just RFID cards.

Steve: Yeah, RFID cards. The guys at F-Secure, as I mentioned at the top of the show, tell the story that one of them had a laptop stolen from a hotel about a decade ago. And in their own FAQ about this they asked themselves, why did the research take over a decade to complete? And they said: "Figuring out the complexities of how the lock system, software, and keys worked was very complicated. Building and breaking an electronic access system is very difficult because there are many facets to get right.

"Assa Abloy [is the name of the company, A-S-S-A, second word A-B-L-O-Y] is a highly reputed lock manufacturer" - actually they're the largest in the world - "and aside from the seemingly innocuous security oversights in the software" - okay, so F-Secure is characterizing these as "seemingly innocuous security oversights" - they wrote "...their products are well designed. These security oversights were not gaping obvious holes." They wrote, "It took a thorough understanding of the design of the whole system to be able to identify small flaws in the system. The researchers then creatively combined these flaws to produce the attack."

So, okay. So what happened? An RFID card is something which is externally powered. Basically it's a...

FR. ROBERT: It's a transceiver.

Steve: ...a transponder.

FR. ROBERT: It's a transceiver, yeah.

Steve: Yeah, exactly. So you ping it with energy, and it gives you back a code. And we know that these particular keys were the "hold them up against the reader" style. They weren't the mag stripe "insert and pull out quickly or swipe." And apparently they are in millions of hotels worldwide. They're called the VingCard, and the software behind it is Vision. And so they simulated the attack with an ordinary electronic key to target the facility. So basically they got one of these keys and built their own RFID reader and pinged it.

And again, they didn't articulate the nature of the attack over time. My guess is that by getting a large number of cards over time - that is, not just one card. They did say that the cards could be retired. They could be obsolete. They could have been, you know, how many people end up walking away and forgetting to turn their little card in. I mean, they're meant to be cheap. Sometimes they'll say "Drop into any mailbox, and it'll be returned to the hotel." So they're not free. But if you forget to return it, no biggie. Well, what you end up with is a key that is some sort of something, basically probably a static serial number which that card has. I'm guessing that it is relatively good encryption. That is, that if you looked at one, you get a large bit stream out.

FR. ROBERT: A blob, yup.

Steve: Yes. You look at another, and you get a very different-looking bit stream out. It's not as simple as, oh, look, these 64 bits are not changing, and these 64 bits are. No. It's like, oh, all 128 bits are different on this one, than this one, than this one, than this one. But probably, if they took enough samples of the static bit streams, they were able to reverse engineer the algorithm which underlay probably the sequential generation of these keys, where in fact there was like a facility bit stream that did not change which all the locks in the entire hotel knew about, and then there was an instance bit stream which

somewhere was an incremented counter. And those two were then mixed together and encrypted to create something which was not at all obvious to a casual observer.

But if you took enough of them together, and if you were F-Secure, and you had a burr in your bonnet over having had a laptop stolen a decade ago, and you just liked a good challenge, you were probably collecting these keys from a given facility over time.

FR. ROBERT: I've got probably about 100, 150 of them. Because they all have the logo of the hotel. That's kind of how I can go back and remember the hotels I've stayed at.

Steve: Yup.

FR. ROBERT: The thing about this that makes it more interesting, from the enterprise side now, most of these locks are not networked. It's not as if they're connected to a master system. They are standalone. And the way that they know that it's time to have a new authentication is because the first time you get your card, added to that is the command to reset. So it's reset to this code. Which means, if you want to update, you have to go lock to lock to lock to lock. There is no way to do a central update. Now, the more expensive ones do have that, but they are very rarely used at hotels.

Steve: Right, right. And one of the things we know is that - so imagine that our model is a - it probably isn't 64-bit facility ID because that's too many bits for the facility. So say it's a 32-bit facility ID. That's how many IPs we have in the world; right? That's 4.3 billion. That's enough hotels. So a 32-bit facility ID. And say maybe a 96-bit serial number. Now, imagine that that is synchronized by the network down in the hotel's management and is a monotonically increasing value. So there's a counter which increments every time a new card is programmed. And the resulting 32 plus 96, so 128 bits, is encrypted - and that's not hard, that's AES, any little microcontroller can do that now - with a secret key. So the facility number, the 96-bit serial number that'll last forever, is encrypted with a secret to produce 128 bits which are, on their face, absolutely random. So that gets programmed into the card.

Now, every door lock, as you said, Padre, they're not networked. They're not talking to each other. But the door lock has the same little microcontroller chip and the secret key. And so when it pings the card, it runs the decryption, and it gets the facility number and the sequence number. So first of all, it checks to make sure, again, assuming this is how this works - I'm making this up because there's no technical information provided, but this is probably something like this. It verifies the proper facility. And when that's done, it then looks at the serial number. And its rule is never accept a lower serial number than the most recent one I've received. And don't accept one that is ridiculously higher than the most recent one I've received.

And so what that does is, that allows it to notch itself forward such that, as soon as a new guest uses the key they've just been issued, it automatically renders the previous guest's key inoperable, which is exactly what you want. You don't want any previous guest to be able to reuse their key. So as soon as that lock sniffs a new card, it goes, oh, look, that serial number has now been moved up by, what, maybe 400, because you could have had 400 keys issued since that room's previous guest had their key issued. But it's not going to be 20 billion. That's a ridiculous high number.

So the point is there's a limit to how far in the future it will go, but it will never go into the past. Which means that the instant it sees a new key, it obsoletes all previous keys and now honors that one until, again, it sees one that is reasonably close in the future. And so that allows a non-network set of locks to be secure in the hotel deployment model. And I would imagine it's something like that. Who knows what the actual bit

lengths were. But these guys probably got a bunch of them, and basically they reverse-engineered the secret key, that is, that key that encrypts the bit stream from the RFID tag. And then, once they had that, they could make their own keys.

FR. ROBERT: Right. I think what we've learned here is that you want to keep everything in the hotel safe because I've learned from Defcon that these hotel safes are absolutely secure. Here he's putting a code of 1234567 - uh-oh, wait. Oh, huh. Yeah, by the way, the hotel safe inside your room, they're junk. They're garbage. They will keep out no one. Just FYI.

Steve: Wow. Yeah, so if you have anything valuable, take it downstairs and make it...

FR. ROBERT: Take it with you.

Steve: Yeah, take it with you or make it the problem of the hotel management and say, look, you need to put this in your hotel safe. I'll come back and get it. Otherwise, no.

FR. ROBERT: It's funny because those hotel - I always assume everyone can see everything that's in my room in a hotel. I've spent way too much in hotels over the last 20 years.

Steve: You have to assume that. Yes, I mean, that is the takeaway for our listeners is you have to assume that we don't have security.

FR. ROBERT: I've put drop cams in my hotel rooms before. And at minimum room service is coming, or the cleaning lady is coming through. So again, just be smart. We're a smart audience, so trust no one, including trust no one who might come into your hotel room.

Steve: I have a nice story, email that I received on April 27th from a - looks like he would pronounce his name Lachlan Gabb. He's in Sydney, Australia. And of course the subject is "Another SpinRite Success Story." He said: "Hi, Steve." This is a nice one. He said: "Hi, Steve. Just wanted to share another SpinRite success story. I'm a security analyst living in Sydney, Australia, and also the go-to guy for computer problems within my family and close neighbors. A couple of weeks ago, one of my neighbors called me to take a look at their computer that had been running very slowly recently. Before I had even arrived to take a look at the system, it had blue-screened and was now refusing to boot at all, stating that no operating system could be found. I asked, but predictably there was no backup, and the computer contained important photos of their grandchildren and documents dating back over five years.

"I removed the drive and connected it to my computer to see if I could recover any data; however, Windows would not recognize the drive at all. Being an avid Security Now! listener I immediately thought of SpinRite, which I had been looking for an excuse to try out for a couple of months anyway. I went over and purchased a copy" - thank you very much - "and let it run on the drive overnight. I think we know how the rest of this goes. In the morning, not only could I copy all the data from the drive, but it would even boot again.

"I promptly copied the data to a fresh drive and reinstalled it in the computer, along with a much-needed lecture on the importance of backups." He says, "I was given immense gratitude and even a container of freshly baked cookies for saving the system." He says, "I wanted to pass on the gratitude to you, but I ate the cookies myself. Thanks for the wonderful product, and I'm looking forward to the next episode of Security Now!. Kind

regards from Down Under, Lachlan." And, wow, thanks for the great story.

FR. ROBERT: Use SpinRite, get cookies. I mean, I don't think we could be any more clear on that.

Steve: Pays for itself in cookies.

FR. ROBERT: And, you know, it's just one of these things that's automatically in my toolkit. I may not need it that often, but when I need it, I have no other tool like it. That's my little pitch.

Steve: So a bit of errata before we talk about Microsoft. Last week you and I talked about Azure Sphere. And I misstated the openness of the system. And it's funny because I read it, but I didn't listen to what I was reading, apparently, because, well, because it is a little confusing. On their slide, which they present as a nice-looking PDF graphic that we linked to in the show notes, it says: "Azure Sphere gives you choice." And I thought, whoa, good. You can connect data from the cloud, proprietary or public, or even to your on-prem infrastructure, to the Azure Sphere security service. Okay, well, it turns out upon closer inspection they have the little arrows pointing around in different directions on this graphic. And you're able to add data and telemetry is what can be connected to non-Microsoft services.

FR. ROBERT: Ah, okay, that makes sense.

Steve: Yes, it does. But the use of Microsoft's, quote, "Azure Sphere Security Service," which is inseparable from the rest, appears to be also inseparable as a component of their system.

FR. ROBERT: See, that makes more sense because I was wondering about that last week. I was like, wait a minute. But if you're not using Azure, how do you get the benefits of the security package? Because that's baked into Azure. That's not a hardware thing, or a software component you can download.

Steve: And you could argue, too, and I do, and I think it makes sense, I wrote here in this errata, "and assuming that they price it all affordably, this offloads ALL [in caps] of the responsibility of maintaining the security and integrity of these IoT devices from hardware manufacturers who just want to crank out hardware without investing in huge aftermarket maintenance and security."

So we would all feel more comfortable if our light bulbs that were Azure Sphere-secured had a connection that was authenticated from a trusted environment in the chip, all the way to Microsoft's secured Azure Sphere security service that was monitoring it and maintaining it and updating it, not to random Chinese company whose employee set it up and quit a year ago and is no longer doing any actual security management. So, yes, I wanted to correct the record. A friend of mine said, "Steve, I think you read that wrong." And so thank you, John, for the correction. I appreciate that.

FR. ROBERT: We actually talked a little bit about this on This Week in Enterprise Tech a while back.

Steve: Ah, cool.

FR. ROBERT: Microsoft has dropped, dramatically dropped the opening tier, so that the opening tier of their IoT services are now way less expensive than they used to be. And

essentially it's a very low monthly fee per device, and then you get a certain number of transactions.

Steve: Ah, okay.

FR. ROBERT: Right. Now, the nice thing about this is they've designed it right. So the big problem with IoT is not being able to design the devices to be robust enough to do what they need to do and also provide security. So instead the devices, they don't accept any queries. They only connect back to the Azure Services. That's what you query to get data from the IoT devices. And that's why the security is then taken care of by Microsoft because you don't have to secure the IoT devices. They're not accessible to anything.

Steve: Well, and you know, that's nice, too, because I'm always nervous about the Internet needing to be able to connect into my network. I don't want UPnP opening holes through my firewall.

FR. ROBERT: Ever.

Steve: Ever, exactly. So it's way better if all of the junk, connected things, if they're making outbound NAT traversal connections to a central service, and its security is strong because it does have incoming connectivity back through the NAT as a consequence of the devices having initiated the outbound hole punching through the NAT, and so then you query the Azure service to find out what's going on. So, I mean, this all does feel a lot better.

FR. ROBERT: It is a lot better.

Steve: And I would love it to succeed.

FR. ROBERT: Now, though, on the contrary, if that main Azure service gets owned, then they can own all the IoT devices connected to it. But that is so much less likely than someone owning millions of IoT devices, which has already happened.

Steve: Has already happened, yes.

FR. ROBERT: All right.

Steve: Okay. So I wanted to wrap up this week by talking about an initiative, I think is probably the way to describe it, and if anyone's interested I have a link to the PDF in the show notes. Microsoft calls it TCPS, Trusted Cyber-Physical Systems. And so this is their moniker for - and cyber-physical is hyphenated. The idea being sort of a catchall for things, like the problems we have now with water treatment plants hooked to the Internet because somebody thought that was a good idea. And, well, it does allow remote monitoring and blah blah blah. But, yes, unfortunately it requires that there be no security mistakes. And exactly as we were just talking about, even if this were to occur with Azure Sphere, it might be that light bulbs, consumer light bulbs have more security than nuclear power plants. Which could be a problem.

FR. ROBERT: Makes total sense. I mean, my light bulbs are essential. Nuclear power plants are really optional.

Steve: Yeah. So essentially, and I'm glad Microsoft has put this out there because it sort of feels bureaucratic, and it's not a spec. Whereas Azure Sphere is actual bits and bolts and a developer board that will be available in a couple of months, and an open design

for a multiprocessor, multicore-based chip which could actually be in light bulbs. This is not there yet. This is sort of the so-called TCPS. This Trusted Cyber-Physical Systems is a forward-looking, I think it was like a 16-page, I think it was 18 pages, actually, 18-page document describing, like, noting the problems that we've had before, securing the Internet-connected valves in important facilities and keeping them out of hackers' hands.

And what this amounts to is them extending the model we already have of security, all the way to the I/O pins on microcontrollers, which is really what they're proposing. We already talked about this notion of a Trusted Execution Environment, that's an acronym, TEE, that they use in this document. A Trusted Execution Environment meaning that, from the moment it is booted, everything it does is bootstrapped, signed, and signature verified before it runs. And there is then no way for a hacker to intercept the process, to get their own code in. And nothing that runs code will run code which is unsigned.

In fact, I think I've mentioned it on this podcast, a few months ago the work on my SQRL client for Windows got to the point where it was time to think about an installer. And it doesn't need one. None of my, as we know, you just run my stuff. But it's meant to be used by regular users, and they're going to click on a download link, and it's going to go into some download directory somewhere which isn't a permanent resting place, or shouldn't be, for anything. So I thought, okay, you know, I have to hold their hand and kind of install. And it may well be that a problem is found after release. That's generally the case, certainly for a security thing. So I need an upgrade facility.

So there's still only one EXE. It's still, and we're at about release point, by the way, where it's 265K because, yes, it's all assembler. And so you just run it. And when you run it, it notices that it hasn't been installed, so it installs itself, meaning that it puts itself where programs are supposed to be and then runs from there from now on. The point of this is that what about updates? What I do is what you have to do if you're truly going to be concerned about security. We've run across so-called "supply chain flaws" in the past where, for example, the CCleaner servers were compromised a few months ago, and lots of people downloaded a malicious version of CCleaner. Well, that can't happen to GRC or SQRL because, if the existing SQRL client learns that an update is available, it downloads it into a temporary location as a non-executable named file, and then it itself verifies the Authenticode signature and that it was not just correctly signed, but that it was signed by me. And only if all of that passes does it then move and rename it as an executable and then arrange to have itself replaced with that.

So again, it's maybe overkill, but I'm sure we'll be glad that nobody, even if GRC's servers were compromised with a bogus version, no SQRL client out there would accept that. So this is that kind of Trusted Execution Environment thinking that we need. And so essentially what Microsoft has done is they've laid out an architecture and also a way of getting from what they call "brown field" to "green field," brown field meaning just the way things are today where you've got compromisable devices which are not protecting their communications. They're not protecting their code, their communications, or the execution microcontroller that is overseeing the state of the valve being open or closed.

And Microsoft lays out the process of - and again, they say right at the top, we're not inventing anything here. We're just saying this is how this has to be done. So let's start thinking in terms of moving the concept of a trusted execution environment all the way down to the field, the so-called now being green field, where at every stage of the process we are only running code which we know is safe to run. And even the microcontroller is protecting itself if it gets updated from a malicious update, and only signed commands where the commands to open or close a critical valve are properly signed through a chain of trust.

And again, they also fully developed this notion of a mature chain of trust so that you have a trust anchor root somewhere. And very much like we have a chain of certificates, there is a verifiable chain where any action chains back to a root that can be trusted. And so long as you protect that, you are thwarting bad guys and knowing, essentially verifiably knowing that the security of your industrial control system is solid, not just hoping that it is so.

FR. ROBERT: Now, you know, Steve, I don't think this is overkill at all. I think this is exactly what is required because, unfortunately, in the era of SCADA, Supervisory Control and Data Acquisition devices, security was tacked on a generation or two after the devices had been created. It just didn't work. And what we'd seen in some of the most recent industrial IoT incidents is not only are they trying to gain access to the equipment, but they also try to erase the logs. That's one of the biggest things right now because they realize it may take them a long time to get the access correct. And in the meantime they want to make sure no one catches onto them so they delete the logs. This handles all of that.

Steve: And it's got blockchain, woohoo.

FR. ROBERT: Yeah. So, I mean, even if you're not able to keep people from probing your network, at least you'll know that they're doing it. And that's actually a huge jump in security. I want to see stuff like this on everything, in this sense. There's been a really big push to secure, oh, well, that pump is connected to the water cooling system to a nuclear reactor. This has to be on every valve. We've also seen trials where people will gain access to a device far downstream or far upstream, but they'll use that to the effect of damaging one of the main devices. That's already happened in Illinois. So this is a very, very good step. And the way that you describe it, it looks like this is very thorough. This is soup to nuts. This is ground-up security.

Steve: Yes.

FR. ROBERT: And, yeah, I mean, I'd like to have this on my IoT devices.

Steve: Well, and we have now all of the crypto tools we need. We now have little chips with sufficient processing power. Once upon a time we didn't have that. The microcontroller was already straining just to do its job. Now it's like how many ARM cores does it have? It's ridiculous how much power we have. So it's wrong not to use the power there which we can to wrap crypto around everything, which is now just not a problem. We've got bandwidth. We've got high-speed connections. Whereas once upon a time we were sending a DC signal - high means open, low means closed.

FR. ROBERT: That's right.

Steve: Now we can send megabits. We can send data. So let's do that. I mean, let's wrap this around, wrap it in a cone of silence, in a cone of crypto, so that bad guys are just locked out from the beginning. And I'm glad you pointed out the secure logging because that is a key part of this that I forgot to talk about, and it's important.

FR. ROBERT: It really is. Well, you know what, I think this might be one of the first times since I've started co-hosting with you that we can end the show on a positive note. That was actually a good story. That's a hopeful story, Steve.

Steve: Let's just hope it spreads.

FR. ROBERT: Indeed, indeed. I want to thank you very much for allowing me to do the show with you. Leo is back next week. And unless he goes on an unscheduled vacation or needs to step away for a while, this may be the last Security Now! I host with you for quite a while. It's been an absolute pleasure, my friend.

Steve: It has indeed, and I really appreciate your standing in for Leo and being here with me for it. It's been great, always.

FR. ROBERT: Indeed. Folks, that does it for this episode of Security Now!. Now, don't forget that we're live every Tuesday on the TWiT TV network at 13:30 Pacific time. Steve will always be here to inject you with a sense of healthy paranoia. Yes, folks, it's healthy because you do need to know the threats out there, and you do need to trust no one, and you do need to listen to our own personal security guru, Steve Gibson.

You can find all of our shows at the show page at TWiT.tv/sn, all the way back to Episode 1, including the show notes and a place to subscribe if you want to make sure that you get all the security goodness in your device of choice every week. We've got an audio version, a video version, and a high-definition video version, all for your pleasure. You can also find Security Now! anywhere that fine podcasts are aggregated, as well as on the GRC website, where you'll be able to download high-quality audio versions. That's also where you're going to find Steve's wonderful products, like ShieldsUP!, SpinRite, and coming soon, SQRL.

Steve, did I miss anything? Do you want to throw any more plugs in here before we sign off?

Steve: No, you've got it good. Everybody knows that Elaine does transcripts for us every week, so we've got transcripts of the podcast, as well. And it's been great, Padre. Thank you so much for standing in for Leo.

FR. ROBERT: Thank you. And in your honor, I will be going back to Windows 3.1. I'm Father Robert Ballecer, the Digital Jesuit, just reminding you that, if you want to keep your data well into the future, you need to think about Security Now!.