TPM v1.2 vs 2.0

Description: This week we look at a way of protecting ourselves from Razer mouse-like local elevation of privilege attacks. We reexamine the meaning of the phrase "Internet Anonymity" following the ProtonMail revelation. We revisit Apple's now delayed CSAM plans. We look at some new troubles for Bluetooth and at a popular and persistently unpatched residential security system which can be trivially disarmed by bad guys. We share some interesting closing-the-loop feedback and a new Sci-Fi discovery. Then we take a long and careful look at the details and differences between version 1.2 and 2.0 of the Trusted Platform Module specification to discover just what it is that Microsoft wants to insist is available for Windows 11.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-835.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-835-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Some big stories. ProtonMail reveals, oh, yeah, after all we do in fact log IP addresses, and we will hand them over to the authorities. Bit of a shocker there. A new flaw in Bluetooth that affects pretty much everybody and everything. And then Steve's going to break down the difference between TPM 1.2 and 2.0 and finally answer the question, why is Microsoft requiring it in Windows 11? It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 835, recorded Tuesday, September 7th, 2021: TPM v1.2 vs 2.0.

It's time for Security Now!, the show where we cover your security, your privacy, everything you want to know about how the Internet works with this guy right here, our Explainer in Chief, Mr. Steve Gibson. Hello, Steve.

Steve Gibson: Yo, Leo. Great to be with you again. We are first episode of September, which is, being Tuesday, this is the - when we have the 14th, next week, that'll be the latest possible Patch Tuesday occurring in the month. And it happens that this is a month anniversary for me. I've been married to Lorrie for five months.

Leo: Happy Anniversary.

Steve: I woke up this morning, and fortunately I remembered to say "Happy Anniversary."

Leo: I think on the fifth month it's not exactly critical. But you'd better remember the first year anniversary.

Steve: I would have survived, yes.

Leo: Yes.

Steve: Okay. We've been talking for the last couple weeks about maybe sort of more the politics of this whole Windows 10 versus 11, 11 needing Trusted Platform Module 2.0. And I thought, you know, our listeners want some deep dive technology stuff. So today is TPM 1.2 vs 2.0.

Leo: Oh, good.

Steve: What is the difference? What is the deal? What is TPM bringing? Why does Microsoft appear to feel they need 2.0 for Windows 11, even though 1.2 works fine for Windows 10? What's the deal?

Leo: What was that sound? You can't let that go.

Steve: That's one of my many various...

Leo: What does that one mean? I have never heard that one, as far as I know.

Steve: And it's a good thing you've never heard that one. That one was GRC's server just crashed and restarted itself.

Leo: Do you want to take a look and push some buttons?

Steve: Oh, no. We're already back up. We're in good shape.

Leo: Well, that's good. In 17 years I've only heard that once, today, just now. So that's good.

Steve: Yeah. I have some very old technology which I know what's going on. I know what the problem is. And I'm going to have to go back into it when I'm getting it ready to handle the betas of 6.1, when people who own 6 want to start playing with 6.1 while I'm still working on packaging up into its finished form. And so in order to make that happen, I'm going to have to tweak the automatic downloading system. And I know exactly the first thing I'm going to do when I open up the server code is to remove a bunch of crap which it involved sort of the dance we did where we briefly moved people over to SSL, back when that was brief, when they wanted to use ShieldsUP!.

In order to avoid proxies, I would establish an SSL connection to them in order to obtain their actual IP and then allow that to be dropped in order to switch them back to normal HTTP to run ShieldsUP! because of course back once upon a time the world was mostly HTTP. Well, obviously that's not the case any longer. But it wasn't hurting anything except that there is this weird edge case with the later version of IIS which doesn't handle some of that correctly. Like again, very unlikely, but every so often it happens. It doesn't hurt anything. Everything's fine. And basically that's my notice telling me that GRC's net engine just restarted. Well, the only reason it restarted was that it stopped, and it had to go, you know, ouch.

Leo: At least it did that automatically. That's good.

Steve: Yeah.

Leo: I love that sound. What is the sound?

Steve: And, see, you love it. It just brings a chill to me.

Leo: Yeah, because you know what it means.

Steve: Exactly, exactly. So it's some horrible siren sound. There was a time about, boy, what was it now, it was maybe six months ago where it was happening often, and I had to stop working on SpinRite in order to address it finally. And I'd forgotten. I told the whole gang in the newsgroup what was going on. But there was something that I had tracked down, I don't even remember now what it was, that was that cause. But now there's this other lingering one. And I'll go and look. I have like a debug trail, and everything's being captured and snapshotted. And so after the podcast I'll go look, and I know exactly where it will be in the code. And it's like, yeah, okay, fine.

But again, it would take me probably - I just don't want to do it until I'm back in the net engine code to implement the changes for SpinRite 6.1 support. So I'm just, you know, it doesn't - it happened, god, I don't remember when. Actually I have a log because it actually sends me a text message. So that's a text message sent by my code to my phone, and then I have that associated with a particular sound on my iPhone.

Leo: Oh, so the phone made the [mimics sound].

Steve: Yeah. It was the phone itself, yeah.

Leo: Cool.

Steve: Anyway, we're going to talk about TPM 1.2 vs 2.0 at the technical level, like what are the functions it offers, what are the functions of it that Windows uses, and why the difference? But first we're going to look at a way of protecting ourselves from those Razer mouse-like local elevation of privilege attacks. We're going to reexamine the meaning of the phrase "Internet anonymity," following the ProtonMail revelation. We revisit Apple's now-delayed CSAM plans. We're going to look at some new troubles for

Bluetooth and at a popular and persistently unpatched residential security system which can be trivially disarmed remotely by bad guys. Not what you want in your residential alarm system.

We share some interesting closing-the-loop feedback. And as I mentioned before we began recording, I wanted to make sure that JammerB, John, knew this, a sci-fi discovery that I just happened upon, I think it was yesterday, when Amazon was showing me some other things I might be interested in. And I thought, what? And I didn't know that something had happened that had. Then we're going to take a long and careful look at the details and differences between v1.2 and 2.0 of the Trusted Platform Module. So I think a great podcast for our listeners.

Leo: Lots to talk about, as always.

Steve: And Leo, if they don't like it, they can have their money back.

Leo: No one has yet asked for their money back.

Steve: No.

Leo: So that's the good news. Well, you had plenty of time to hydrate, Steve, and prepare for...

Steve: Thank you for that.

Leo: ...the Picture of the Week.

Steve: Which is not very inspired. But it just seems apropos.

Leo: It's apropos, yes.

Steve: Yes, of our topic. For those who are not looking at video, it's just your screen that has a dialog titled "Windows 11 Setup." And it is declaring this PC can't run Windows 11. And then there's a little "Here's why" underneath. And it says: "The PC must support TPM 2.0."

Leo: Aha.

Steve: Yeah. And we know lots of people have been seeing that, and it's been annoying. We're going to be talking about exactly why this is the case at the end of the podcast.

Leo: Yay. Yay.

Steve: But first, two weeks back, we covered the industry's discovery that the tongue-in-cheek phrase "plug and pray" was more apropos - there's that word again - than we imagined, with the news that plugging a Razer mouse or keyboard into any PC could be used to trivially and locally bypass that system's possibly carefully crafted user privilege restrictions to give the operator full system-level rights. The worry was that, since this was such a simple mistake to make, that is, for the developers of the mouse driver installation software to make, that it must certainly have also been made elsewhere, and no one had just sort of noticed.

You know, it's like one of those things, you know, like suddenly people, well, it's like when you buy a new car, and suddenly that's all the car that you see on the road. It's like everyone seems to have that. How did I not notice that before? Similarly, now that we have this new thing that we've realized may be pervasive, people are starting to look for it. And wouldn't you know it, indeed, other similar instances are being discovered.

Okay, now, the good news is that Will Dormann at CERT's Coordination Center has found a registry key that governs Windows' ability to automatically download and run any of those not-explicitly-asked-for-and-executed plug-and-play peripheral installation packages. Last Tuesday, a week ago, Will tweeted: "Do you not like the fact that connecting a Razer device auto-runs an arbitrary installer with privileges? Do you suspect that other devices may be exploitable? Fear not." And then he gives a "set" and then a long-looking path to the key in his tweet.

BleepingComputer posted, and I grabbed from there, thanks to them, a snapshot of the registry with that key open. Now, okay. I'll tell you what the key is in a minute. But let's remember that I'm the guy who left the DNS IP address override in his ecommerce server's HOSTS file, causing its attempted connections to fail when our ecommerce provider's IP address changed, as it certainly had every right to. In other words, I forgot that I did that, and it came back to bite me. So this suggests that if I were to make such a change to prevent the auto download and installation of drivers when I plug things into my computer, I might...

Leo: In three years we're going to do a show where you say Windows Plug and Play never works.

Steve: It's crap, Leo. I'm plugging this thing in all day, and nothing is ever happening.

Leo: This is probably not the perfect fix for this.

Steve: Well, yes. What you'd like would be a little pop-up dialog where you would say, "Hi, we'd love to install this for you, but you need admin rights for that to happen. Got any?" Anyway, so instead it just, like, nothing happens. So the point being that we both just made to be clear about it is that we might forget that we had disabled this automatic device driver download and installation.

Leo: Which is really hideously useful. I mean, let's face it.

Steve: Yes. Yes, yes, yes.

Leo: We count on that. How many times have you plugged in a USB device and heard "bing-bong," and then, you know, "boom-boom" as it installs, and you...

Steve: Right.

Leo: But you do usually see a little pop-up in the registry.

Steve: Well, yeah. And if you're able to click on your little installing device driver thing down in the tray, you'll see this little whatever it is, spinning or doing something, it's like, we're polishing your Windows. And maybe you'll be able to see something pretty soon like your icons. That would be nice. Anyway, so I'm just saying, okay. For anyone, probably those with enterprise IT management responsibilities, who is interested, the details are in the show notes. And for anyone who's comfortable with the Windows registry, there's nothing to it. I mean, it's simple to do. You go to HKEY_LOCAL_MACHINE, then SOFTWARE > Microsoft > Windows > CurrentVersion > Device Installer. Now, on my machine, that key existed, but it had no names and values associated with it. It was blank. But it was nice to see it was there. So you'll be able to find it.

Once you're there you create a REG_DWORD value named "DisableCoInstallers." If you want to do this all by audio of the podcast, with the D of Disable, the Co of Co, the C capitalized, and Installers capitalized. I don't remember whether registry names are case-sensitive. They may not be. Values tend to be, but maybe not the names. Anyway, DisableCoInstallers as all one string with D, C, and I capitalized, and set its value to 1. Which, as its name suggests, disables coinstallers. And now, probably after you reboot, when you plug something in, your system will no longer run off to go download it and run it.

So again, the danger is that someone will forget that this was done and be puzzled when things don't work automagically as expected. But in many settings that might be exactly what's desired. So I did want to share with our listeners who may be responsible for machines whose users cannot be fully trusted that this option is there. And I agree with you, Leo, be nice...

Leo: Lot of businesses you don't want plug-and-play running anyway.

Steve: Unh-unh, no.

Leo: Because somebody could have a malicious USB key. There's all sorts of side effects, yeah.

Steve: Yeah. So I wish that the phrase "Internet anonymity" were not an oxymoron. Any longtime follower of this podcast would have seen example after example, because we've covered them in the last 16-plus years, of the truth that while the Internet can make tracking people much more difficult, it's still not impossible.

This is why over the years we've developed the model of having two people who really wish to hold a private conversation strip down to their shorts and walk out into the middle of Central Park with no one around, carrying a large and thick quilted comforter blanket. They then throw the blanket over themselves, then whisper as quietly as

possible, so that they're barely audible, into each other's ear in order to exchange secrets. That's about as low-tech as anything could possibly be, which is exactly the point. Technology provides vast measures of convenience, but that convenience always comes at the cost of some bit of security. Maybe not a lot. Maybe a lot. Who knows? Anyway, in the doghouse this week is ProtonMail.

Leo: Oh, yeah.

Steve: And I know a lot of our listeners are ProtonMail users. My Twitter feed is full of people saying, "Hey, talk about ProtonMail, Steve. Tell us how wonderful it is." And "I'm using it, and I think it's great." So, okay. What's really cool, Leo...

Leo: You know what I've said to everybody, and I'm vindicated now. Email is inherently insecure, period.

Steve: Yes.

Leo: Don't kid yourself.

Steve: What's cool is they got caught, and I've got it in the show notes, changing their website. So they boast on their website secure email based in Switzerland. But that statement's meaning was changed last week. I just scrolled down through their home page at ProtonMail.com. And, oh, boy. It looks wonderful. The Hacker News...

Leo: You made fun of it when it came out because you mentioned the Swiss server in the mountain. Remember that?

Steve: Yes, yes. Yeah. So the Hacker News writes that: "On its website, ProtonMail advertises that: 'No personal information is required to create your secure email account. By default, we do not keep any IP logs which can be linked to your anonymous email account. Your privacy comes first.'" Unquote from them; unquote from the Hacker News. But when I just went to that page and searched for bits of that assertion, that statement, it had apparently been removed. I wonder why? Could it perhaps be because it was recently discovered that ProtonMail had provided location information to Swiss authorities which directly led to the arrest of one or more of the users of their supposedly Swiss identity-protecting service? Yeah, that might have been a factor.

The Hacker News captured a "before" screenshot of ProtonMail's homepage. Right at the top it lists its main feature categories as: "Swiss Privacy," "End-to-End Encryption," and "Anonymous Email." But it doesn't say that today. Their homepage has been changed. Today, the top line still begins with "Swiss Privacy" and "End-to-End Encryption." But now the third item is "Your Data, Your Rules" where it previously said "Anonymous Email."

Now, I give them some serious props for 'fessing up and apparently thinking, okay, we really can't make that Anonymous Email claim anymore, can we. You know. That had to hurt. Because I know these guys' heart is in the right place.

Leo: I feel bad for the French climate activist who believed them and is now in jail.

Steve: Yup. ProtonMail acknowledged that it had received a "legally binding order from the Swiss Federal Department of Justice" related to a collective called Youth for Climate, which it was "obligated to comply with," compelling it to hand over the IP address and all information it had related to the type of device used by the group to access the ProtonMail account. Hmm. That's probably not the nature of the protection that those users of ProtonMail believed they were receiving after reading ProtonMail's original, quite powerful and compelling privacy-expounding home page - which, as I've noted, no longer exists.

So despite its no IP logging claims, the company acknowledged that, while it's illegal for the company to comply with requests from non-Swiss law enforcement authorities, it will be required to do so if Swiss agencies agree to assist foreign services such as Europol in Europol's investigations. In part of a lengthy response posted on Reddit, the company said: "There was no possibility to appeal or fight this particular request because an act contrary to Swiss law did in fact take place, and this was also the final determination of the Federal Department of Justice which does a legal review of the case."

Put simply, ProtonMail will not only have to comply with Swiss government orders, it will be forced to hand over relevant data when individuals use the service to engage in activities that are deemed illegal in the country. This includes monitoring its users' IP addresses. ProtonMail's founder and CEO Andy Yen tweeted: "Proton must comply with Swiss law. As soon as a crime is committed, privacy protections can be suspended, and we're required by Swiss law to answer requests from Swiss authorities. It's deplorable" - this is his tweet. "It's deplorable that legal tools for serious crimes are being used in this way. But by law we must comply with Swiss criminal investigations. This is obviously not done by default, but only if legally forced."

Leo: I have to say that seems to me disingenuous because they were saying "We don't log IP addresses. We don't preserve that information." If they had received a warrant for information they didn't have, they wouldn't have to hand it over. So they were misrepresenting what they were doing.

Steve: Well, unless the warrant said...

Leo: Start logging, yeah.

Steve: Yes, exactly. So...

Leo: Now, at that point Swiss law requires they notify the subject.

Steve: Ah.

Leo: So that's again another question mark. At that point they're supposed to tell the subject we are now monitoring your login. Did they do that? You know, this is never - and this is independent of whether they're encrypting email or not.

Steve: Yeah, yeah, yeah, agreed.

Leo: Of course they're encrypting it.

Steve: Oh, yeah. Absolutely end to end. We believe that.

Leo: End to end if the other person is going to do it with you. I mean, you know...

Steve: Right.

Leo: You can't send an encrypted email across the public Internet. You can only send a link saying there's an encrypted file for you. Or, you know, you can come unencrypt it, something like that.

Steve: Sure.

Leo: Because Gmail doesn't know how ProtonMail works. Gmail wouldn't know what to do with it.

Steve: What I'm wondering, you know, because, I mean, they're loudly talking about this service. The entire reason people use ProtonMail is to get the protection that, you know, they used to claim "anonymous email." So, I mean, it does seem to me like maybe their technology's not working right, but that their heart's in the right place. Their business is to offer, to the degree they can, absolute privacy. But they're certainly correct in explaining that, if they have the ability to comply with such requests, they must by law do so.

Leo: They have no choice. Yeah. Same with Apple. Same with everybody else.

Steve: Yes. Exactly. Exactly where I'm going with this, Leo. What I wonder is whether it wouldn't be possible for them to design a system where it's just not possible for them to comply, no matter how much they might be forced to. And then for example, to the point you just made, we conjectured here that one possible motivation for Apple's quite unpopular plan to compare photo signatures against a known photo signature database before uploading new photos to iCloud might actually be so that they could further lock down iCloud as they have their other end-to-end encrypted systems. In other words, make it impossible for them to comply to such a request. And where there's a will, there's a way.

So ProtonMail users who are concerned about the visibility of their IP addresses I think should also take the trouble to route their access through the Tor network to obtain additional anonymity. Since email is not real-time communications anyway, routing email through Tor, which adds significant communications latency on its own because you've got to bounce through all these nodes and have onion wrappers unwrapped, to me that makes a lot of sense. But presumably that isn't something that these guys thought they had to do because they were being told that their IP address was not being logged.

We don't know the details of whether they were told, you know, forced to start logging, or whether maybe they actually were logging and deleting them, quietly keeping a log, but never intending to publish it. Or maybe they understood that they did have an obligation to comply with Swiss law and were actually logging. We don't know.

Leo: We don't know.

Steve: But speaking of Apple's client-side image matching, for those listeners who don't already know, Apple has announced that it will be pausing the planned rollout of its quite controversial plan, which as we know screens users' devices for child sexual abuse material, so-called CSAM. As we know, Apple received substantial and sustained pushback, not only over worries that the technology could be weaponized for mass surveillance and erode the privacy of their users, but just over the "ick" factor, at the idea of having anything to do with this kind of material being preloaded into all iOS devices which are capable of uploading images to iCloud. People just didn't like it.

So what Apple wrote, they said: "Based on feedback from customers, advocacy groups, researchers, and others, we have decided to take additional time over the coming months to collect input and make improvements before releasing these critically important child safety features."

So they're not suggesting that somehow this problem doesn't still need to get solved. As has been noted, other services which are performing this kind of image matching are discovering a horrendous amount of illegal content, which is what this stuff is. So as you guys did on MacBreak Weekly, I give Apple credit for publishing their intentions to develop and use this technology well ahead of their planned incorporation of it into iOS 15, and then listening to and responding to all the blowback that they received. Of course the trouble is from a technology standpoint, you know, it's pretty much binary.

So I wonder what Apple's going to do. Either you do it this way or you don't do it this way. They're saying, well, we're going to review and think about this some more. But if their underlying motivation was actually to somehow arrange to lock down iCloud in an end-to-end encrypted fashion, which we know they haven't yet, they might be able, again, under the "where there's a will, there's a way," they might be able to do that by, as I know Rene had previously suggested, by somehow offloading the image analysis to an intermediate cloud-based service which would perform the signature comparison, and then would encrypt the image on behalf of the image's iOS user for then permanent storage on iCloud.

And this might mean that they need to go back to the drawing board and work a bit harder at coming up with a solution that both matches the needs of their users, which arguably what they were proposing did not due to the outcry, yet still offer, you know, come up with a technologically sound way of doing the image verification that everybody else is doing, while also arranging not to be able to decrypt it once it's been submitted to iCloud.

Leo: So just because I know you'll get emails, I'm going to clarify.

Steve: Ah, good.

Leo: Save you from Twitter storms. At no point were those images ever transmitted. The only place in the country that can legally store those images is NCMEC, the

National Center for Missing and Exploited Children. It was founded by the U.S. government, funded by the U.S. government. It's a nonprofit. But law allows it and only NCMEC to keep those images. NCMEC generates the fingerprint, the neural hash.

Steve: Right.

Leo: That's the only thing that was loaded on your phone is that neural hash.

Steve: Right.

Leo: So Apple's using the process, which was developed by Microsoft, by the way, to compare that neural hash of the image that you have on your phone to that neural hash from NCMEC. So I think everybody does it. Even Apple has been doing it to email attachments. Google does it to Gmail attachments. Google, Dropbox, Facebook, Microsoft all do it on their cloud storage.

Steve: Right.

Leo: I think if Apple had said, "We're going to do it on your cloud storage," that would have been fine. But as you point out, you could not then have end-to-end encrypted cloud storage because Apple would have to access your images. So I think that they were trying to make it still possible to do end-to-end iCloud by putting the fingerprints on your phone, having your phone do the comparison, and then all the phone does at that point is generate a voucher.

Steve: Yup.

Leo: And after 30 vouchers are generated, it then sends the images to Apple for verification, which is a little creepy, and then on to NCMEC to report you. So there's no "ick" factor in a sense that you have those images on your phone. You don't.

Steve: Correct.

Leo: I think the "ick" factor was people don't want Apple to be doing stuff on your phone, scanning your images for anything. Right?

Steve: Right.

Leo: I don't - would anybody object, I mean, we're already - Dropbox does it. Facebook does it. There are [crosstalk].

Steve: Well, and of course the other point was it was only those images which were being submitted to iCloud, also. So if you turned off iCloud...

Leo: Right, you can turn that off.

Steve: ...photo backup, then...

Leo: No problem.

Steve: We're back the way it has been.

Leo: And I would presume that the evil entities that are doing this already know you don't send it as an email attachment. You don't store it on a cloud service. And now they know turn off Apple's iCloud. As if they were turning it on in the first place.

Steve: Right.

Leo: I think it's a little disingenuous of Apple to imply that they're doing anything to stop CSAM. This is a sop to law enforcement so that they could turn on, I think, end-to-end encryption on iCloud.

Steve: Well, yes, because what do we hear is law enforcement brings out the kiddie porn issue whenever the issue of encryption comes up, saying, well, if you're encrypting everything, then this is going to be abused by child pornographers.

Leo: And terrorists, yeah.

Steve: Yes.

Leo: And I think honestly at some point, you've already said this, Apple's going to have to some way give in. Everybody's going to have to some way give in to law enforcement.

Steve: I think that's where we're headed, yeah.

Leo: They're going to be forced to. I mean, Australia, Canada, England. Canada's got a very draconian anti-encryption law about to be enacted. It's going to happen. And if it happens there, it'll happen here eventually.

Steve: Yeah.

Leo: All right. I just wanted to make sure - because I don't want anybody to think that Steve thinks they're putting child pornography on your phone. He doesn't think that.

Steve: Right. And we covered this extensively, of course, over the last two weeks before.

Leo: Yeah, yeah, that's how I know it.

Steve: So if I misspoke, I'm glad you grabbed that and corrected it, Leo. Thank you. Oh, Leo, Bluetooth has new troubles.

Leo: Oh, yes, saw this. Oh, god.

Steve: God. A new set of attacks are known as "BrakTooth," "brak" being Norwegian for crash. Paraphrasing from the discoverer's disclosure, they explain. They said: "Bluetooth Classic protocol is a widely used wireless protocol in laptops, handheld devices, and audio devices. In the past few years, Bluetooth has come under scrutiny due to the discovery of several critical vulnerabilities. In this report, we disclose BrakTooth, a family of new security vulnerabilities in commercial Bluetooth stacks that range from denial of service via firmware crashes and deadlocks in commodity hardware to arbitrary code execution in certain IoTs," you know, Internet of Things.

"As of today," they wrote, "we have evaluated 13 Bluetooth devices, meaning core chip things, not actual products, 13 BT devices from 11 vendors. We have discovered a total of 16 new security vulnerabilities, with 20 common vulnerability exposures (CVEs) already assigned, and four vulnerabilities are pending CVE assignment from Intel and Qualcomm.

"All the vulnerabilities are already reported to the respective vendors, with several vulnerabilities already patched" - that's of course, you know, from now on out - "and the rest being in the process of replication and patching. Moreover, four of the BrakTooth vulnerabilities have received bug bounties from Espressif System and Xiaomi. An exploration of Bluetooth listings reveals that BrakTooth affects over 1,400 product listings. BrakTooth exposes fundamental attack vectors in the closed" - and I'll get to that in a second - "Bluetooth stack.

"As the Bluetooth stack is often shared across many products, it is highly probable that many other products, beyond the approximately 1,400 entries observed in Bluetooth listings, are affected by BrakTooth. Therefore, we suggest vendors producing Bluetooth system-on-chips (SoCs), Bluetooth modules, or Bluetooth end products use the BrakTooth proof-of-concept code to validate their Bluetooth stack implementation." In other words, you know, basically a fuzzing attack against Bluetooth. They conclude: "The availability of the BrakTooth proof of concept is discussed at the end of this report."

Okay. So here again we have a problem with the way we're still doing technology, like with the idea that we're going to use and trust proprietary voting machines. You know, similar kind of problem. In both the Bluetooth and the voting machine cases, a closed system must be trusted to do the right thing while its not doing the right thing can have sweeping consequences for millions of people. Yet we just shrug. "Oh, darn."

So we already know that opening everything up doesn't by itself automatically make everything secure. So while it's not sufficient, it is almost necessary, I would argue. To my mind, this is the single most compelling reason to push for opening our technologies. And Leo, I know you're on the same channel as I am with this.

So here again with BrakTooth, we have researchers who, because this technology is closed, have been forced to hack into and reverse engineer important implementations of technology that will be used by millions of people. And once shipped and sold, will almost certainly never be updated, once it's in the field. The good news this time is that being Bluetooth, the attack range is short. And that most of the flaws found simply crash or, in their language, Brak the stack. So, okay. That's good. But this is another of those things where there are doubtless many agencies spread around the world who are sucking up these vulnerabilities and may very well have applications for many of them today. How many movies have we seen where somebody "runs a bypass" on a keypad to gain entry into a protected area? That fictional scenario is becoming less and less farfetched with each passing year or month because all this stuff can be hacked.

The trouble is these IoT things will probably never be fixed. Microsoft cannot manage to get their deployed Exchange servers patched. So what chance does the manufacturer of a residential alarm system have? It's got to be near zero. But first they have to want to, and care to. At least we know Microsoft eventually decides, oops, you know, we'd better patch this. So you have to want it in the first place for there to be any chance at all. But do the manufacturers of IoT things care? Why would they? They've already got their customers' money.

We are rapidly filling the world with a bunch of incredibly complex crap that ships the moment it stops crashing in the lab. And that's an entirely different result from having potentially mission-critical technologies being actively resistant to attack and abuse. They're obviously not. The moment anyone starts poking at our stuff, it collapses. So these researchers just showed us once again that the world we're living in today is one where hoping for the best is apparently all we can do. If I sound disgusted, yes, you know, it's true.

Which brings us to our next story. Attackers can now remotely disable Fortress WiFi Home Security Alarms. Two vulnerabilities have been discovered by Rapid7, a well-known security research group, in the Fortress S03 WiFi Home Security System that can be abused by a malicious party to gain unauthorized access with an aim of altering the system's behavior, including disarming the devices without the victim's knowledge. Just because I was curious, Fortress S03, is that around? I put a link in the show notes to Amazon's Fortress Security Store, where there you can find the Fortress S03 and buy it. I don't advise anyone do that, by the way.

The two unpatched issues were discovered and reported, as I said, by Rapid7 back in May. And they honored a 60, actually more than that, let's see, May, June, July, August, yeah, we're in September now. More like they honored a 60-day disclosure deadline and gave Rapid7 an extra 30 days, so more than 90 days. So let me say that again. Fortress was notified of these problems in May. They've done nothing. The Fortress S03 WiFi Home Security System is described as a do-it-yourself, you know, a DIY alarm system where you buy the various pieces and hang it on the wall and stick them on the windows and things, that enables users to secure their homes, kind of, and small businesses, maybe, from burglars, fires, gas leaks, and water leaks by leveraging WiFi and RFID technology for keyless entry. Yeah, very keyless.

According to Fortress's website, its security and surveillance systems are used by "thousands of clients and continued customers." Not me. I hope not by any of our listeners. And if you happen to have one, maybe go shake their tree because they haven't apparently even fixed it themselves, like for new stuff.

Rapid7 describes the vulnerabilities as being "trivially easy to exploit." Now, I should note they waited three months. They just posted full exploit details on their site. So it was trivial before, if you knew how. Now everyone knows how, and it's even more trivial. So they noted that CVE-2021-39276 concerns an unauthenticated API Access - so, you

know, it has to be it's only hard to use if you don't know about it, but if you do, you can because it's unauthenticated - that enables an attacker in possession of a victim's email address to query the API to leak the device's IMEI identity number, which also doubles as its serial number.

Armed with the device's IMEI number and the email address, the adversary can proceed to make a number of unauthorized changes, such as disabling the alarm system via an unauthenticated POST, you know, an HTTP POST request without the owner's knowledge. While this is not usually much of a concern for random, opportunistic home invaders, they write, this is particularly concerning when the attacker already knows the victim, meaning somebody who lives there or a business, who are securing their business with one of these things. And of course it might be an ex-spouse or some other estranged relationship partner, like somebody who the business fired who's annoyed with them.

The second vulnerability, CVE-2021-39277, presents similar problems, but requires less prior knowledge of the victim, as the attacker can simply stake out the property and wait for the victim to use the RF-connected devices within radio range. In other words, these things are sending and receiving information without authentication. The attacker can then replay the "disarm" command at any later time, without the victim's knowledge, and voila, walk right in.

And now that Rapid7 has made their disclosure, which of course as I said includes full details, anyone with a bit of tech savvy can have at it since Fortress apparently isn't concerned and/or may not be able even to field update their devices if they were concerned. But they've evidenced no concern because they haven't patched it, having been given now more than 90 days. So once again, this is the world we're living in today. Hope for the best. Hope is all we apparently have.

Leo: Of course, I doubt that there's a firmware update patch for those little handheld remote devices; right? How would you even update them?

Steve: Right, right.

Leo: So they probably just go, well, oh well. Oh well.

Steve: Yeah. Wish we knew. And apparently they're not bothering to fix them even for new ones being shipped.

Leo: Well, that's what's disappointing. They could fix it in new ones.

Steve: Yeah.

Leo: Except then they'd have to patch everything because - ugh. What a mess.

Steve: Yeah. It is. It is. And as I said, we are filling the world with complicated crap, and - ugh.

Leo: [Growls]

Steve: Yeah, exactly. Okay. So after our discussion last week of our life hanging by a PIN, I got a bunch of great closing-the-loop feedback from our listeners. I'll share three.

Steven - and it's funny. I like this. His Twitter name is Steven. His Twitter handle is @wnevets. And I recognized "evets" because that's Steve backwards. So "nevets" is Steven backwards. And so this is Steven W, apparently, who reversed his name. Now, Steven, do not use this as your password. It's not a good idea. So he tweeted: "I don't know if anyone has suggested this, but I use Google Voice to secure my SMS fallbacks for services that require it. Taking over my Google Voice number requires taking over my Google account, which doesn't have SMS fallback enabled."

Jon tweeted, @JonMcD86: "Hey, Steve. In this week's Security Now! episode you touched on the perils of using SMS messaging to your cell phone for authentication. I just wanted to throw one thing out there that I do that others might find useful. For platforms that require SMS authentication," he wrote, "I like to use a Google Voice phone number to receive text messages that I protect using my personal YubiKey. It's annoying having to sign into this account repeatedly, and it's far from a perfect solution. But I feel like it's better than leaving this, as you put it, backdoor into my services where SMS recovery is a requirement." He said: "Anyway, huge fan. I listen every week. Keep up the great work."

And finally, Ant Lednev, tweeting under the same name: "Hi, Steve. As I am watching Security Now! and where you describe the T-Mobile, SMS, and LastPass with SMS issues, as well as banking requiring SMS, just want to mention that I am using Google Voice for this purpose. It is protected by Google login and two-factor authentication, has an iPhone app, and you can access it from the browser, too. Curious what your thoughts are on replacing cell SMS to VoIP SMS services."

And to everyone, these three and others, I just wanted to share this. Obviously this is a win, to set up a Google Voice account, use the Google Voice number as your SMS. And I guess it was - I think it was Jon who said it's a pain to log into that, but obviously you don't have to often, only when you're expecting to receive a text message in order to perform account recovery, which you have to go to the trouble of logging in and then capture an SMS text that you received at that phone number or that pseudo phone number, and then you'd have it. So anyway, great tip. I wanted to send the feedback from our listeners back to our listeners, for anyone who might find that useful. Leo, do you know, is it free? Is it a completely free service?

Leo: That's a good question.

Steve: I did not look into it.

Leo: It's free. Yeah, you know, because I pay for Fi. But I think they still offer it as free free, yeah.

Steve: Cool.

Leo: Yeah. My Google Voice got replaced when I used that number for Google's Fi service. And now I pay for it. But I think, yeah, I think it's free, yeah.

Steve: Alain (A-L-A-I-N) said: "I've really come to like @Tailscale as my VPN replacement - enough, in fact, to write a review about it." And I then have the link for anyone who's interested in the show notes. He says: "Thanks @SGgrc for mentioning it." And so it's very - just the top of it is very brief. I thought I would share this real-world experience.

He wrote: "First off, I've tried configuring WireGuard multiple times, around six by my count, on different devices - an Ubuntu VM, a few Raspberry PIs, and a pfSense firewall - all with various versions of success. None of my attempts ended up working fully, though. I would be able to access internal resources, but not the Internet; the Internet, but nothing internal; or nothing at all." He says: "By this point there would be a Raspberry Pi-shaped hole in the window." Meaning, obviously, he threw his Raspberry Pi out the window and left a Raspberry Pi-shaped hole. He said: "So I gave up." And then he says: "Bad tech guy. Go sit in the corner and use a Windows machine for a while."

"Then," he writes, "during my weekly listen of Security Now!, Steve Gibson mentioned Tailscale as a service that uses WireGuard (Episode 830), and that it's really easy to configure. I figured, why not? Let's give it a go." And he says, "...and I like it! The highlights: Works on Ubuntu, or any Linux distro for that matter. Works on Mac/iOS devices. Works on Windows. Yay, I guess. Works on Android. It's fast." He says: "The free tier is very generous." He says: "The paid versions aren't bad either, especially 'Subnet router failover' on the Business plan!!" So anyway, just wanted to share one last bit of happiness.

And Walt Stoneburner tweeted: "Steve, enjoyed your last show's quick dive on assembly language and shout-out to SoftICE." And that was actually, Leo, you prompted me to expand on my talking a little bit about my use of assembler. And he says of SoftICE: "Oh, how I miss that. While I hope to high heaven that someday you write," meaning me, "an assembly language book that goes into your own..."

Leo: He's shaking his head no, for those not watching.

Steve: Unh-unh, that will not happen, "...into your own personal thoughts and techniques..."

Leo: You'd sell five copies. Come on.

Steve: Yeah, "...based on years of real-world usage..."

Leo: It would be [crosstalk].

Steve: Yeah, well, "...are there any resources you'd recommend over any others for experienced C programmers?" Okay. So I replied to Walt, recommending that if he's able to find anything written by Michael Abrash on eBay or in any used bookstore online or offline, that he would never regret spending a few bucks on anything that Michael Abrash ever took the time to write. I own, and they're actually - let's see. How can I - they're right there.

Leo: Right above his head, folks.

Steve: Yes. "The Zen of Assembly Language, Volume 1: Knowledge." And Michael obviously had some ambitions for other volumes. There were never any others under that title. But we got "The Zen of Code Optimization," "The Zen of Graphics Programming," and "Michael Abrash's Graphics Programming Black Book," which is a big monstrous collection of his various articles. Now, they were written back in the days when memory and CPU cycles were both ultra scarce and highly prized. I mean, this is the code that allowed Doom and Quake especially, actually Michael was one of the authors of Quake...

Leo: John Carmack, who wrote Doom, said he based a lot of his stuff, including the Mode Y video mode, on Abrash's work.

Steve: Yeah.

Leo: He was a big fan.

Steve: And sadly, those days have passed. I'm still writing that way myself, not because it makes any practical sense, but because I apparently love moving known-size data, and not very large data, around between registers and memory. I was thinking about this. The best analogy, because I keep solving the same problem over and over and over, and I love it every time, it would be to a craftsman who loves to build homes by hand, loves measuring each piece of timber, planing its surface to make it smooth, and fitting the perfectly sized pieces together, pounding each nail in separately. And once that's done, standing back, taking a slow deep breath and taking quiet personal pride in a really nice piece of work. And it doesn't matter whether anyone else appreciates it or even ever sees it. It wasn't done for anyone else. It's something you did for yourself, for the sheer joy of using your hands to solve a collection of little puzzles. I've never found anything as satisfying as coding.

Leo: I agree, 100%.

Steve: Electronics for me is up there, too. I really do like problem-solving with an environment that has constraints, you know, like how do I solve this problem with these pieces? But anyone who loves coding for its own sake in any language might google the phrase "programming gems." That's sort of become the catchphrase for the art form side of coding. Coding and algorithms are complex enough that particularly elegant solutions often exist.

So I recall, back at the start of my implementation of SQRL, I needed to quickly look up a user-interface string by its index. So I needed a dictionary that I could access at very high speed. The way to do that is with a binary search. And though I'd already implemented countless binary searches through the years, the planets that day must have been in perfect alignment because I wrote the most elegant binary search for 32-bit Intel assembler I had ever written in my life. Its use of registers, condition codes, and conditional branching, I mean, it was just perfect. And, yeah, I know. Talk about hyper-geeky. But that's where I live.

Leo: That's very satisfying, though. There's something just incredibly satisfying when you've got something like that. Are you saying "gems," or do you mean "Programming Pearls," the Jon Bentley series?

Steve: Well, I think that's a little specific. If you do google "programming gems," you'll find a bunch of stuff.

Leo: The problem is unfortunately Ruby now calls its libraries Gems. And so there's going to be collision there. But, yeah. The "Programming Pearls" book is exactly what you describe, which is a bunch of kind of clever algorithmic solutions that make you go, oh, ooh, ah.

Steve: In that case, yes, I would say "Programming Pearls," too.

Leo: It's a classic.

Steve: That's a good tip.

Leo: I was looking for a book, one of Abrash's books. His "Zen of Assembly Language Programming" is available used on Amazon for $488. It's a collector because it's 30 years old.

Steve: Yeah. It's beautiful. Actually I pulled it down off the shelf...

Leo: It's long out of print.

Steve: I pulled it off the shelf this morning, just to look around through it.

Leo: Nice.

Steve: Yeah.

Leo: Yeah, I don't think people write assembly language books much anymore.

Steve: No.

Leo: Don't think there's a huge market for it. Apparently the last job he took in 2014 was at Oculus, doing VR.

Steve: That's where he is, yeah, yeah. He's like chief scientist at Oculus.

Leo: Isn't that awesome? Yeah.

Steve: Yeah. And I'm sure he's trying to bring the latency down because that makes everyone nauseous.

Leo: Exactly why he's there.

Steve: Oculus rhymes with nauseous. One final tweet. Steve Yates tweeted: "Thanks, @SGgrc. InitDisk has just rescued a 64GB USB key that failed to respond to anything else." And I just wanted to take the moment to remind our listeners that this is always the way I operate. Remember that when I began the work on SpinRite, the first thing I did was to solve the problem of absolutely, finally, once and for all, permanently solving the problem of formatting thumb drives because that's the way people would be running with SpinRite, no longer putting it on a floppy and then booting a floppy.

And it turns out it's a lot tougher to do that than you'd expect, which is why for example nothing else would format Steve Yates' USB key other than InitDisk. I did once and for all solve that problem. And so InitDisk exists, and many people discover that it's their tool of last resort that allows them to bring a drive back to life, a thumb drive that just appears to be dead. InitDisk will do it. So just a reminder about that.

And now, believe it or not, Peter Hamilton has been involved in the creation of a short book. The title is "Light Chaser," and it was co-authored by Peter F. Hamilton and Gareth L. Powell. It was quietly released two weeks ago, on August 24th. And I wouldn't have been aware of it if I hadn't been looking for something else when Amazon brought it to my attention.

Leo: See, those recommendations do work.

Steve: Uh-huh. I thought the same thing.

Leo: The first one.

Steve: The Nerd Daily had this short little blurb to say. The Nerd Daily wrote: "Peter F. Hamilton and Gareth L. Powell team up in this explosive, action-packed novella about a love that transcends lifetimes and is powerful enough to destroy an empire. Amahle is a traveler who sails through the universe with nothing for company but the ship's AI. Known throughout the universe as a 'light chaser,' her route takes her to worlds throughout The Domain, where she collects memories in exchange for various goods. But she discovers memories from different lives and different worlds that are meant for her and seem to be from the same person. She begins to question her entire existence. Each memory..." and blah blah blah.

What I liked about it is that it says: "For a shorter book, this story packs an incredible punch. 'Light Chaser' comes in at 173 pages, yet it is epic and expansive, taking us through worlds and lifetimes in rapid succession. Rather than feeling rushed or lacking, the prose is razor sharp, carving out exactly what we need to understand the world and the technology while propelling us forward." And I should just mention I read about five or six reviews. They all fall on that point. They talk about how it is fast, but compelling.

And what made me think of it, of course, is you and I, Leo, have talked about how sometimes we get lost in Hamilton's writing.

Leo: There's a lot of details, yeah.

Steve: With like, blah, blah blah, blah blah. And it's like, okay, well, maybe it's interesting that he found one of his pants legs to be too long that morning. But okay, why do I have to know that? So this might sort of be like ultimate Hamilton, where you just get the good bits, and you don't have to go to the tailor in order to make sure that your hems are the same, because we really don't need to know that. What we do really need to know is what we're going to talk about next.

Okay. So let's begin with a quick discussion about why any of this TPM stuff is needed at all. The presumption is that once an operating system has taken possession of a system's hardware, configured and placed its processors into protected mode, there's no possible way for anything to happen on the system that the operating system cannot intercept and consider. In other words, programs running under such an operating system are supposed to be 100% absolutely contained as an OS client. They're unable to do anything that the operating system does not explicitly allow. Any unallowed action, such as attempting to touch the system's underlying hardware, or read or write outside of fixed boundaries, will raise an exception condition. This causes the offending client instruction to be suspended before it's able to take its action, and returns control to the operating system for consideration of what to do.

Now, this might be something benign, such as a client attempting to access some valid memory that it does have access to, but which has been swapped out of the system's RAM to its virtual memory backing store. In such a case, the OS will schedule the RAM to be retrieved for its client and will briefly suspend the client until its instruction that triggered the memory exception can be restarted, and then it will succeed.

So in an ideal world, once it's started, the operating system has all the tools it needs to actively supervise, manage, and control all of the subsequent activity occurring on the system. And for the most part this works. As we know, it's still an imperfect system because operating systems have become incredibly complex, and human programmers make mistakes. As a result, the designers of malicious software keep finding ways to circumvent this control to elevate the privileges of clever clients that they design. This allows their malicious clients to obtain the same access rights and privileges as the underlying operating system, thus completely escaping the operating system's attempts to control and thereby obtain free rein over the system.

So the ambition of truly secure computing continues to face the challenge of maintaining control over its deliberately misbehaving malicious clients once it's in control. But this assumes that the operating system has not somehow been first maliciously modified, either while it's in cold storage, before it's been booted, or before it's able to finish booting and assume that control over the system's processors and other hardware. In other words, it's becoming quite difficult to attack an operating system once it has control of the system because, mistakes notwithstanding, human error which occurs, modern processor technology can provide absolute control. But what protects the operating system itself from being maliciously modified before it has the opportunity to begin protecting itself?

The answer is the Trusted Platform Module or, more generically, some form of Hardware Root of Trust. This is shortened, popularized, and simply referred to as "Secure Boot." Security Now! Episode 500, which we recorded a little over six years ago on March 24th, 2015, was titled "Windows Secure Boot." So I'll refer any of our listeners who want more

details about exactly how this works step-by-step to that podcast. And all of that remains entirely relevant today. And we talked about all the various keys that are used. Step by step, we broke down the entire process of going from a motherboard with a Trusted Platform Module that had some secrets embedded in it all the way up to Windows operating and all the stages in between.

But the simple fact is secrets are required for security. And the protocols which are used to verify those secrets must be tamperproof. There must be some things an attacker does not know, cannot know, and cannot manipulate. This will prevent an attacker from subverting the operation of an otherwise completely open and publicly known system. For example, we're only able to believe that we're connecting to the proper remote Internet service because the certificate it presents contains a valid signature that's been signed by the secret private key maintained by a third party whom we trust to have verified the attested identity of the remote service. Or another example: When two agencies wish to conduct a private conversation in plain sight, they use a key agreement protocol to derive a temporary secret key which they'll then use to encrypt their subsequent communications.

No matter how open any system is, there must be someplace where secrets can be privately manipulated so that no one, not even someone having complete access and visibility into the system, can interfere with the system's intended operation. In the case of a personal computer, where an attacker, or an attacker's code, might have access to the system before the operating system has booted, there must be a little black box of some kind which is beyond any attacker's ability to interfere or manipulate. The Trusted Computing Group has carefully defined a minimal set of features and functions to standardize the operation of such a black box and has named it the Trusted Platform Module.

As suggested by the title of this podcast and the controversy surrounding Microsoft's determination to raise the version requirements of a system's TPM from 1.2 to 2.0, there have been and are two major release versions of the Trusted Platform Module. TPM 1.2 was first released 16 years ago, back in 2005, the first year of this podcast, and it received its final revision in 2011. TPM 2.0 was first released nine years ago, in 2014, and was last tweaked two years ago in 2019. And the reason it's possible to have a first release of v1.2 or 2.0 with subsequent tweaks that don't change the version is that the 1.2 and 2.0 refer to API definitions rather than implementations.

Now, the TPM is a thing. It's often built into a system as a standalone chip soldered onto a laptop or desktop motherboard. The Gigabyte desktop motherboard I'm using has a socket for an optional TPM; and, as we can see from the Amazon screenshot that I put in the show notes, this is a common option. For those who are not looking at video, this shows - I just pulled this up. I think I typed in "TPM module" into Amazon, and up came a bunch of them. So this shows an ASUS TPM for $33.71, a Supermicro TPM for $39, and an ASRock TPM 2 for $29.01. And then another one from ASUS. And I didn't want to spend any more space on the show notes. But this thing, I think it was a screen of one of seven. And so this, like, scrolls on and on with all these modules.

So the historical logic employed by motherboard makers was that even though a TPM has been available for the past 16 years, it's certainly possible that someone might want one, but especially early on, no one really needed or cared. So they were thinking, why burden the cost of a motherboard with something that would sit there unused for the most part? Thus they just had little sockets. And if you actually wanted a TPM, you could buy one aftermarket and plug it into the little socket. And now your motherboard would be equipped with a TPM.

So of course this thinking is obviously changing now for those who want additional security. And note that since a TPM is actually nothing more than a set of callable

functions, it has an API. There's nothing preventing it from being incorporated into a system's core silicon chipset. And that's exactly what Intel has done. Ever since Skylake, which is the 6th-gen systems, nearly all Intel CPUs have an embedded TPM 2.0 which Intel calls Platform Trust Technology, or PTT. AMD CPUs have an embedded TPM 2.0 called fTPM ever since 2016's AM4 platform. fTPM stands for firmware TPM. It's also possible to do the entire TPM in software. Microsoft has a pure software-only simulator. But of course doing so loses all of the true security advantages of having secrets protected with an impenetrable black box of some kind.

Okay. So what is a TPM? The Trusted Computing Group's own white page describing their TPM is dry, incomplete, and states that TPM v1.2 is the latest current version. So someone hasn't been paying much attention to keeping that page up to date. To begin to answer this question, we'll turn to the source of all the controversy, Microsoft. They both helped design and, of course, use the TPM. What they wrote, it's a bit self-aggrandizing, but I want to share it since it provides a few surprising bits and pieces that aren't found elsewhere.

They said: "Microsoft has led the architecture and adoption of the TPM since its inception. Microsoft invented and contributed the attestation, sealing, and Platform Configuration Register (PCR) features to the original TPM, and contributed to the overall design. More recently, Microsoft architected and edited the TPM 2.0 specification. Many new concepts and features were introduced with TPM 2.0, including crypto-agility, easier management, a more flexible authorization model, and better extensibility. TPM 2.0 devices are now available from many vendors and are incorporated into most business-class PCs and many servers. TPM 2.0 is also making increasing inroads into network equipment, mobile, and IoT devices.

"The TPM 2.0 specification is unique in that it is machine readable. Most of the normative behavioral specification is written in a subset of the C programming language, and the TPM programming interface is defined in machine-readable tables. This allows vendors to quickly build high-quality and interoperable TPM implementations. The TPM is a low-cost, but powerful and flexible, cryptoprocessor. A TPM does many of the things that a smartcard or hardware security module (HSM) does, For example, it's able to create, manage, and use cryptographic keys, as well as store confidential data. But a TPM is intimately tied into how a computer boots and runs, which means it is far more powerful and useful than a simple 'smartcard on the motherboard.'

"For example, platforms that incorporate TPMs measure" - that's the PCRs - "measure and log the software that boots on the device. The resulting boot log can be used to verify that devices are running known software and are up to date using a TPM feature called 'quoting' or 'attestation.' The boot log can be used to protect keys for disk encryption because the TPM incorporates a feature called 'sealing' that can be used to make sure the encryption key is only disclosed to authorized software, and not to disk-cracking tools. Other advanced TPM features include a secure clock, monotonic counters, meaning counters that will only count upwards, a non-volatile storage facility, and very flexible and secure mechanisms for key management operations like key import and export."

Okay. So we have a bit of an overview. Most of the articles describing TPM 1.2 vs 2.0 get stuck on the fact that 2.0, being newer, adds support for some additional newer and stronger cryptographic primitives. Specifically, the original TPM 1.2 builds in functions for SHA-1; HMAC-160, which is HMAC based on SHA-1; 1024 and 2048-bit RSA public key ciphers. To those functions, TPM 2.0 adds the 256-bit flavors of SHA and HMAC, so SHA-2 256 and HMAC-256. 2.0 also adds several 256-bit elliptic curve ciphers and elliptic curve Diffie-Hellman.

Since TPM 2.0 is backward compatible with TPM 1.2, software could simply continue using the TPM 1.2 functions, even if you had TPM 2.0, to run on platforms having either TPM standard. And this is, of course, what Windows 10 does today and will continue to do for the next four years until October 14th, 2025, which is the targeted end-of-support date for Windows 10. And we may see the extended support for cost, which Microsoft is currently doing with Windows 7.

But Microsoft may have decided that it's time to force a change. We've seen and discussed countless examples of this throughout the life of the podcast. It's no longer possible to connect to remote web servers over SSL 2, and usually not over SSL 3. And when I go to Ivan Ristic's terrific SSL Labs facility to ask it about GRC.com, it says: "This server supports TLS 1.0 and TLS 1.1. Grade capped to B." So you can't support anything older than TLS v1.2 if you want to get an "A" from Ivan. I don't. I would rather support older protocols because that's better for GRC, and I can live with Ivan's "B."

And we know that forcing a change in protocols just for the sake of forcing a change, when the currently used protocols appear to be working just fine, is always going to be painful. So if Microsoft wants to force their Secure Boot and other TPM-dependent technologies forward with Windows 11, they'll want to drop support for the older SHA-1 and HMAC-160 and insist upon having Windows 11 only rely upon and use the newer 256-bit crypto technologies. And of course there are other new crypto technologies in TPM 2.0 that are not part of 1.2.

But there may be three things, or there are three things, that Microsoft may have missed. The first is that dropping old and adding newer crypto technologies has traditionally been a matter of upgrading a server or client's software. If your browser doesn't support the latest protocol, you click on "upgrade." But in the case of TPM, the cryptography is locked into the hardware. By design, it cannot be upgraded because it must be designed to be an inviolate black box. So while it's easy for Microsoft to say, "We're going to require the use of 256-bit crypto technologies which are missing from TPM 1.2 and are present only in TPM 2.0," not only is it not easy, it's not possible for Windows users to fix this with a software upgrade.

The second thing they've missed is the lack of any compelling reason to move from Windows 10 to Windows 11. During last week's Windows Weekly, Mary Jo expressed her puzzlement about the entire thing, asking Paul whether there was anything truly new about Windows 11 beyond and she really asked this its centered Start Menu and its rounded corners.

Leo: It might have been me that asked that. But it's definitely a good question.

Steve: You guys have all been asking. And this was Mary Jo Foley, who would know, if anyone in the world would; right? She would normally be all over it. Paul in this instance shrugged and joked that there were, after all, quite a few corners that had needed to be rounded during the creation of Windows 11. So, yeah, you know, they did a lot of work there.

And the third thing Microsoft may have missed is that many people using Windows 10 today are not using any features of TPM, regardless of edition. They aren't booting with Secure Boot enabled. They don't use BitLocker. They don't even have TPM enabled. They don't want or care about those things. So the security profile of these people suggests that TPM doesn't matter to them at all. Yet Microsoft is planning on denying these people the option of moving to Windows 11 for no reason other than that features they are not using, and presumably have no interest in ever using, are absent from their hardware. No nicely rounded corners for them.

So Microsoft appears to be saying: "We've decided to force a change to Windows that no one really needs because we want to force people to use more modern hardware even if they don't need more modern hardware." This doesn't feel like a winning strategy to me. When I was introducing TPM 2.0, I said that most of the articles describing TPM 1.2 vs 2.0 get stuck on the fact that 2.0, being newer, adds support for some additional newer and stronger cryptographic primitives. The point I was making is that there's more to 2.0 than just stronger and newer crypto.

Okay. So first I'll enumerate the five functions that TPM 1.2 offers, then the six additional features which the architects of TPM, among whom Windows or Microsoft clearly feels that they number, added in the move from 1.2 to 2.0. Okay. So TPM 1.2 provides for identification of devices. Prior to the release of the TPM spec, devices were mostly identified by MAC addresses or IP addresses, not security identifiers. Okay. So that tells us something we have not really paid attention to. There is a unique serial number of some sort in the TPM which software is able to query.

Second, secure generation of keys. Having a hardware random number generator is a big advantage when creating keys. We know about that. A number of security solutions have been broken due to poor key generation. And actually we covered this at the time on the podcast. The Infineon implementation of TPM 1.2 actually had a problem with its RSA key generation, and they needed to do a big mea culpa and fix that, and did. Also, the secure storage of keys is the third feature of 1.2, like of all TPM. Keeping good keys secure, particularly from software attacks, is a big advantage that the TPM design brings to a device.

Fourth, NVRAM. When an IT organization acquires a new device, it often wipes the hard disk and rewrites the disk with the organization's standard load. Having nonvolatile RAM allows a TPM to maintain a certificate store, meaning across that wipe. And, finally, device health attestation. Prior to systems having TPMs, IT organizations used software to attest to system health. But if a system was compromised, it might report it was healthy - of course it would - even when it wasn't.

Okay. Those things exist in all TPMs. With the passage of time and the benefit of experience with TPM 1.2, TPM 2.0 then added the following: algorithm agility. Algorithms can be changed without revisiting and revising the specification, should they prove to be cryptographically weaker than expected. So that's cool. We don't need to change the spec anymore. We can just extend its algorithms. Enhanced authorization. This new capability unifies the way all entities in a TPM are authorized, while extending the TPM's ability to enable authorization policies that allow for multifactor and multiuser authentication. Additional management functions are also included. So this sort of feels like something Microsoft had a hand in, in 2.0.

Quick key loading. Loading keys into a TPM used to take a relatively long time. They now can be loaded quickly, using symmetric rather than asymmetric encryption. As we know, we talked about this, asymmetric encryption is actually incredibly slow. It's cool what it does, but it does not do it quickly. Therefore nobody uses asymmetric encryption to bulk encrypt anything. What everybody does is use a high-quality pseudo random number generator, or hopefully a true hardware random number generator, to get a large, completely random key. You then use asymmetric encryption only to encrypt it. And then you use the symmetric key with a lightning-fast symmetric cipher to do the actual bulk encryption. So anyway, 2.0 makes key loading way quicker.

Also what they describe as non-brittle PCRs, those are the Platform Configuration Registers. In the past, locking keys to device states caused management problems. Often, when a device state had to go through an authorized state change, keys had to be changed, as well. This is no longer the case. So they just created some additional architectural stages, a little more architectural complexity to decouple some things.

Also, flexible management. Different kinds of authorization can be separated, allowing for much more flexible management of TPM resources. Again, that sounds like something Microsoft probably pushed as a result of experience with the TPM 1.2. And, finally, identifying resources by name. Indirect references in the TPM 1.2 design led to security challenges. Those have been fixed by using cryptographically secure names for all TPM resources.

Okay. So those last things are the things that were added to the 2.0 spec on top of 1.2. And for anyone who's interested, I found a fabulous reference. I've got links to its PDF and its EPUB version. It's titled "A Practical Guide to TPM 2.0." It's over on the Springer Link site. Again, link's in the show notes. Fabulous reference.

Okay. So Microsoft provides us a chart of their technologies which require TPM 2.0 and which will not run under 1.2. Actually, it provides two columns showing us which ones run everywhere and which only run under 2.0. I've got a link in the show notes. And Leo, thank you. You brought the chart up on the podcast. I'm going to run through these to give our listeners a sense for what we're seeing. Basically, what we're seeing is 14 items. The TPM 2.0 column is all checked in green because all 14 run with it. In the TPM 1.2 column, we're missing four features out of 14. The other 10, they work just fine under 1.2.

The so-called Measured Boot instruments the booting process, storing the history in the TPM to detect future tampering. Microsoft says that it runs fine under TPM 1.2 today, right now. Microsoft says that BitLocker runs equally well under 1.2 and 2.0. However, they explain that Device Encryption requires TPM 2.0, and that it's not available with 1.2. Okay, well, device encryption is actually kind of watered-down BitLocker. Since BitLocker, which is actually superior, runs with either of the TPMs, unless you're a Windows Home user without BitLocker the lack of device encryption is a "who cares." The only place it is important is if you're a Home user, you will have device encryption, so you get the watered down version of BitLocker, if that matters to you.

The Windows Defender Application Control runs under either TPM. Microsoft explains what that's about. For Microsoft Edge, this Application Guard helps to isolate enterprise-defined untrusted sites, protecting, they say, your company while your employees browse the Internet. As an enterprise administrator, you define what's among trusted websites, cloud resources, and internal networks. Everything not on your, basically it's a whitelist, is considered untrusted. If an employee goes to an untrusted site through Microsoft Edge, the site is opened in an isolated Hyper-V-enabled container. And what I just described runs fine under both TPMs. Doesn't care.

Now, something called Windows Defender System Guard is a post-boot integrity verifier which works with the TPM to verify that Windows wasn't compromised during the boot. That's sort of like it augments Secure Boot, and it does need 2.0. Not available if all you've got is TPM 1.2, although Secure Boot is available.

Seven other security features - Credential Guard, Device Health Attestation, Windows Hello, UEFI Secure Boot, TPM Platform Crypto Provider Key Storage Provider, Virtual Smart Card, and Certificate Storage - all work fine with either TPM 1.2 or 2.0. No difference.

Something called Windows Autopilot does require 2.0. I had to look that one up. Microsoft describes it as a collection of technologies used to set up and preconfigure new devices, getting them ready for productive use. Windows Autopilot can be used to deploy Windows PCs or HoloLens 2 devices. Okay. So maybe not a big loss if we don't have that under 2.0. It's 2.0 only. And finally, SecureBIO, also known as Microsoft Enhanced Sign-in, requires TPM 2.0. This is an enterprise-targeted biometric identity system. Now, that's

the case, even though Windows Hello, which provides a lot of the same features, also works just fine under only TPM 1.2.

Okay. So in conclusion, there are no obvious showstopping technologies present in TPM 2 that we cannot readily live without if all we have is TPM 1.2. If all we have is 1.2, we get BitLocker, but not the watered-down, feature-stripped Device Encryption. We don't get the enterprise double-check-after-boot System Guard, but no one running Windows outside of an enterprise would have either of those anyway. We also don't get Autopilot or SecureBIO, even though we do get fully functional Windows Hello. In other words, while TPM 2.0 might enable a few extra features for the enterprise, TPM 1.2 provides everything that typical Windows users need, which is what we have today.

And now, as I put all this together, what I finally realized was the following: Everything I've just enumerated all refers to the way everything is today under Windows 10. Nothing changes under Windows 11. In other words, there is not a single new feature being brought to Windows by Windows 11 that creates any additional need or reason for 2.0. When Mary Jo asked Paul what was new in Windows 11 aside from the centered Start Menu and the rounded corners, she was really asking. There are no new features in Windows 11 that require anything more of the TPM than Windows 10 already does. Yet Windows 11 is refusing to run on the same TPMs as Windows 10, apparently because someone at Microsoft thought it would be cool to enact a more restrictive change in requirements.

Given these realities, the path Microsoft, I think, should take for Windows 11 is clear. Simply use the maximum security that's being offered by whatever, if any, TPM is present in a system. If the platform offers TPM 2.0, great. Use the 256-bit enhanced security that's available there. If not, settle for the 160-bit security offered by SHA-1 and TPM 1.2's HMAC, just as Windows 10 does now. If a platform doesn't offer TPM 2.0, then its user cannot take advantage of those four enterprise-oriented features from among the 14, among the balance of 10, that will run on any TPM.

So, fine. Explain to those enterprise users that if they want those four features, they'll need to upgrade their hardware. But don't tell random home or small business users who couldn't care less about Windows Defender System Guard and Autopilot that they're SOL if they wish to upgrade to the new Windows, just because it's possible to do so, to be mean. That's how it's going to be seen. It's going to be seen as capricious and arbitrary because, as we've just seen, it is.

It's my sincere hope that this decision within Microsoft is being given a broader airing, and that it's a requirement that they might reconsider. We just saw Apple reconsider on the CSAM deal when people said, whoa, wait a minute. Given a careful analysis of the two TPM feature sets, and of Windows' current use of them, there doesn't appear to be any compelling need to force a move to TPM 2.0 "just because." Oh, and after I wrote this, I just found out this morning, Windows 11 Dark Mode will have more smoothing sounds.

Leo: Soothing sounds or smoothing sounds?

Steve: Soothing. They're going to be more soothing, Leo.

Leo: Oh.

Steve: When you switch to dark mode, you will be further soothed.

Leo: Well, that's a relief.

Steve: Maybe you will be soothed by Microsoft changing their policy because this is nuts. I mean, again, maybe they're saying we want more security. Maybe they're saying we have plans in the future. Okay, fine. So tell people if they want to use these new things they need to upgrade their hardware. But don't prevent them from having what, I mean, everybody's going to want 11, only because everybody always wants the latest and greatest. Well, of course that wasn't really true for Windows 10. But Microsoft is going to be pushing it. So let people have it. There is no reason not to. Windows 10 is going to continue to run for the rest of its life. It's crazy.

Leo: I'm glad their sounds are more soothing, though.

Steve: Won't that be nice, yes. You will switch to Dark Mode on Windows 11 for more smoothing sounds.

Leo: So we really - this really flattens it.

Steve: It does.

Leo: There's nothing in the 8th-generation Intel processors that makes them more secure.

Steve: No.

Leo: And there's nothing in TPM 2.0 that is necessary for Windows 11, at least its current...

Steve: There's nothing that 11 is using.

Leo: Right.

Steve: There's nothing it's using.

Leo: I could see if, oh, you need it for BitLocker, but...

Steve: We already have it.

Leo: We already have it.

Steve: BitLocker runs on TPM 1.2, yup.

Leo: So it's to sell new computers, which is what we always thought it was, to be honest. In fact, this confirms it. Good analysis.

Steve: It's not going to go down well, my friend.

Leo: Yeah. Yeah. They just want you to buy a new computer. It's good for the industry. Steve Gibson's good for the industry. I can tell you that right now. We thank you so much for listening to Security Now!. If you want a copy of this show, you can get it of course from us, but Steve's got copies. In fact, he's got some unique versions of this show on his website. He's got the 16Kb audio for people who want the smallest possible audio file. Even smaller, the human-written transcripts, which you can read along as you listen, you can use for search. It's a really nice feature. All of that is at GRC.com, the Gibson Research Corporation.

While you're there, pick up a copy of the world's best mass storage maintenance and recovery utility, SpinRite. Current version 6.0; 6.1 is in process. You'll get a free upgrade, plus you get to participate in the development of 6.1. There's also a lot of other great free stuff at GRC.com. You can leave him questions there at GRC.com/feedback. But his DMs are open on Twitter. That might be the better way to do it. He is @SGgrc. Actually, he also has some great forums, where there's always good conversation going on at GRC.com.

We have 64Kb audio and video versions of the file at our site, TWiT.tv/sn, so you can download it. You can actually watch us do it live. If you're really in a hurry, you want to get it sooner than anybody else on the planet, tune in every Tuesday right after MacBreak Weekly. That's usually around 1:30 Pacific, 4:30 Eastern, 20:30 UTC. Live audio and video streams at TWiT.tv/live. If you're watching live, chat with us live. There's two places to do that. Everybody can chat at irc.twit.tv.

Let's see, what else? Oh, you know what, if you want the easiest way to get this show, just find a podcast application and subscribe. You'll get it the minute it's available for download. That's probably the easiest. I do invite you, if you subscribe, and your podcast player has reviews, it'd be nice if you left a five-star review for Steve. That'd make everybody feel extra special. Plus it helps spread the word, and we appreciate that.

All right, Steve. Next Tuesday I don't think we'll be delayed. We might be. Remember the Apple event is at 10:00 a.m., followed by MacBreak Weekly at 11:00. I think Apple's going to be dead on time, which means we'll probably be on time to Security Now!. But just we might be pushed back a little bit, just a word of...

Steve: Is this going to be new phones?

Leo: New phones are here. We think.

Steve: Cool.

Leo: And a new watch.

Steve: Cool.

Leo: But we won't know till Apple...

Steve: And by the way, I did love that Google spoof of...

Leo: Wasn't that great?

Steve: The circle, oh. I grabbed it, and I played it for Lorrie. She just loved it.

Leo: She got it?

Steve: With the whispering voice. Oh, yeah. With the whispering voice. It's an earplug.

Leo: This is in their ad for the new Google Phone, the 5a with 5G, because it has a headphone jack. So they do the Jony Ive-style parody.

Steve: Ah.

Leo: Which if you could find it is well worth it.

Steve: It's really wonderful. I found it on YouTube, so I'm sure it's - yeah. Okay, my friend.

Leo: Thank you, Steve. Have a great week, and we'll see you next time on Security Now!.

Steve: Thanks, buddy. Bye.