ROCA Pain

Description: This week we discuss the inevitable dilution in the value of code signing; a new worrisome cross-site privacy leakage; is Unix embedded in all our motherboards?; the ongoing application spoofing problem; a critical IP address leakage vulnerability in TOR and the pending major v3 upgrade to TOR; a Signal app for ALL our desktops; an embarrassing and revealing glitch in Google Docs; bad behavior by an audio driver installer; a pending RFC for IoT updating; two reactions to Win10 Controlled Folder Access; a bit of miscellany; some closing the loop with our listeners; and, three weeks after the initial ROCA disclosure, I'm reminded of two lines from the movie "Serenity." Assassin: "It's worse than you know." Mal: "It usually is."

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-636.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-636-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Lots to talk about. ROCA, of course, the public key crypto error that is causing some pain all over the place is the topic du jour. But there's a whole lot more, including antiviruses that just aren't doing their job. You'll find out why next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 636, recorded Tuesday, November 7th, 2017: ROCA Pain.

It's time for Security Now!, the show where we cover the latest security news and what to do about it. It's both prescriptive and informative. Mr. Steven "Tiberius" Gibson, our host.

Steve Gibson: Yo, Leo.

Leo: Good to see you, our moustached man.

Steve: Re-moustached.

Leo: Glad it's back, yeah.

Steve: Everybody is.

Leo: All's right with the world.

Steve: Interesting experiment. It's like, okay. Fortunately, it will grow back.

Leo: You know, it's funny. For me, it's the opposite. I grow facial hair, I keep trying to do it, and then I give up after a week because it's itchy, and I don't like it. And so I never get there.

Steve: Yeah. There is definitely an itchy phase you go throw where the hair's kind of poking back in at you. It's like, okay, let's not do that. So Episode 636. And I struggled for a name. There is some additional news, not surprisingly, about ROCA, which we covered three weeks ago after its initial announcement. And it's funny because I was put in mind, I have no idea why this came to me, but it was a line that I liked from one of our favorite movies, of course science fiction, "Serenity."

Leo: Oh, yeah.

Steve: Where the Assassin says to Mal, he says, "It's worse than you know." And Mal says, "It usually is." And such is the case with ROCA. But before that we're going to talk about the inevitable dilution in the value of code signing. A new worrisome cross-site privacy leakage. The press went wild in the last couple days over the concern that we all had an embedded Unix in our motherboards.

Leo: Shocking. Actually, it's MINIX. It's not even Unix, it's MINIX.

Steve: Exactly. It's mini-Unix.

Leo: It cracks me up, yeah.

Steve: And so what does that mean? But what's also cool is for somebody who wants to be on the bleeding edge, and there may be blood involved, someone has put together a how-to on shutting that down.

Leo: This is the management, Intel Management Engine.

Steve: Yes. And it involves solder.

Leo: Oh, dear.

Steve: So anyway, but it's very cool. Then we're going to talk about...

Leo: Solder is the new tinfoil hat, by the way.

Steve: That's right. Solder and a Raspberry Pi 3.

Leo: Oh, geez, Louise.

Steve: It's a perfect thing for Father Robert to do on Know How.

Leo: Oh, good, yeah.

Steve: When you have a motherboard you don't mind never having operate again.

Leo: Right.

Steve: Because the risks are high. But the rewards, it's just sort of a really fun hacking adventure. The ongoing application spoofing problem, which recently hit Google's Play Store with more than a million downloads of a bogus WhatsApp app. A critical IP address leakage vulnerability in Tor, which of course is the whole reason you use Tor is not to have your IP address leak, so whoopsie. Then we also have the pending major v3 upgrade to Tor, which has been four years in the making, and it's now in alpha. So we'll talk about that.

And I'm very excited that there's now a Signal app for all of our desktops, which puts some interesting pressure on iMessage because I'm chafing at the fact that, as a Windows user, I don't have any access to iMessage on the platform I'm sitting in front of. Mac people do, of course. But Signal now cross-desktop. There's also, and I heard you talking about it, it must have been with Jeff and Stacey last week, this embarrassing glitch in Google Docs, which surprised a lot of people about how much Google was looking at what they're doing.

Leo: Yeah, no kidding.

Steve: Yes. Also some bad behavior from an audio driver installer. A pending RFC for IoT updating, I mean, if there was anything on our Christmas wish list, that's what it would be. Also two different reactions to Windows 10 Controlled Folder Access that we talked about last week. A bit of miscellany. If we have any time left, we will have some feedback from our users before we get to, yes, it's worse than you know, and it usually is, talking about ROCA.

Leo: I love it.

Steve: So I think another great podcast.

Leo: Well, I can't wait, as they say. All right.

Steve: So our Picture of the Week, which we are not going to skip, as I forgot to last week...

Leo: Well, we ended up showing it.

Steve: Yes, thank you for reminding me. It's like, Steve, did you forget something? And this is interesting in the context of everything we've been talking about relative to the vulnerability of the devices which are facing the Internet. And someone sent this to me, and I thought it was just perfect. It's a notice sent from Charter Communications noting that TWC is now Spectrum. And the title is "Important Information Regarding Your Internet/Voice Modem."

"Dear Valued Customer," it reads. "As part of our continued effort to give you the best service experience possible, we will be performing an upgrade to your modem. We will begin updates starting November 6, 2017. Updates will occur between the hours of midnight and 6:00 a.m., Monday to Friday, during a four- to six-week period." So that sounds like they're going to be pushing patches, firmware, to the entire customer base during the early morning hours so that it's less intrusive.

They say: "During this upgrade, your modem will go through a series of steps and will reboot. As a result of the reboot, your modem will be offline for two to four minutes. During this time you will not be able to access the Internet, receive incoming calls, or be able to dial 911 and 611, and your local and long-distance residential voice service will be temporarily disabled. Your modem should not be powered off or disconnected during this upgrade as this could cause your equipment upgrade to not occur." Yeah, no kidding. Sorry, we're not able to install new firmware if you've unplugged it. "Once complete, your device will automatically reboot. We hope you will enjoy your improved features, security, and more with this upgrade."

So of course they're not saying what this is. But so it's a good sign that they're being proactive. And presumably there are, I mean, to do something like this, which you don't see that often, there is something that they feel they need to fix. So props for them doing it, and looks like it's a pretty big campaign for them.

Okay. So the inevitable dilution of the value of code signing. A report came out in the last week from some researchers, I think it was three, oh, yeah, three University of Maryland at College Park researchers who took a look at the prevalence of signed malware, which unfortunately is no longer an oxymoron.

Leo: I was going to say, that's a paradox. Signed malware?

Steve: Exactly. So, okay. So we know about signatures. You take some blob of anything, and you hash it to create a fingerprint of that blob. And the power of the hash function is that you change even, you know, any one bit change in the source that you're feeding in on average inverts 50 percent of the hash's bits. And there's no way to design an

augmentation or a change to the source that doesn't dramatically change the hash. So essentially the hash is a fingerprint. You then sign the fingerprint in a way so that the validity of the signature can later be verified. So, and we've talked about of course doing this in all different kinds of contexts.

Well, Authenticode is what Microsoft calls their digital signing system. And over time the requirement for that's been growing. For example, it was controversial when Microsoft first introduced the idea that drivers would need to be signed. And of course that's sort of, you could argue, the most critical portion of the system because the driver by its nature is running with full privileges down in the kernel, able to do anything it wants. So you would hope that signing something brings more integrity to it, more trustworthiness. GRC's apps, my stuff, ever since this sort of became important, has been signed. So, for example, Never10 has a digital signature. All of the SQRL betas have digital signatures. I've been signing stuff just because, for example, especially for new software, AV, antivirus software, will use that, whether or not software has a signature, as another signal to it.

So, for example, if Gibson Research Corporation achieves a positive reputation with an AV vendor, and something new comes along that itself doesn't have a reputation, but it's digitally signed by a company that does have a reputation, then by inference you can say, oh, well, we trust them, so we trust what they have signed. So there's sort of the same so of implied trustworthiness downstream from signing.

Now, the problem, though, is that there's a much lower bar for qualification for a code-signing certificate. Basically you raise your hand and you say, "Hi there, CA, Certificate Authority. I'm breathing. And I want to sign code." And they go, "Okay, pay us." And so you pay them some money, and they give you...

Leo: Are you sure you have to be breathing? I mean...

Steve: Actually, bots could probably do it, too.

Leo: Bots can do it, too, yeah.

Steve: Yeah. So unlike the traditional relationship that we've talked about often here with certificates, where it's a DV where you need to prove domain control, or an OV where you need to approve organization control, or an EV, which is the highest level of enhanced verification. This is basically just, "Hi, I want to sign stuff," and you buy a certificate, and you can. So naturally, as soon as AV software, antiviral software began, well, began using signatures as a signal, an additional signal for it to use, malware authors began saying, oh, we should be signing our malware.

So these three guys at the University of Maryland at College Park took a look at 325 signed malware samples. Of those 325, 189, which was 58.2 percent, carried valid digital signatures. Which is to say that somebody issued, apparently the malware authors, or maybe a front for them, a valid signature, I mean, a valid certificate that they could then use to sign their code. Just like I do. I mean, I have a certificate, of course, from DigiCert, and I'm signing - I signed Never10. I sign all the version of SQRL that we're working on. So the problem is it's not difficult to get one.

And so, but what's really interesting and a little disturbing is in their research they found

that, as I said, 189 out of 325, which is just shy of 60 percent, carried valid digital signatures, whereas the balance of those signed malware samples, which is to say 136 of them, carried malformed signatures. That is, anyone actually verifying the signature would find it wasn't any good. It was some blob, some signature taken from other software that had simply been stuck onto the malware. And what they found was that many very prominent antiviral software don't take the time to check the validity of the signature.

Leo: Aw, geez.

Steve: I know.

Leo: What a surprise. These guys are nitwits. Oh, my.

Steve: They're just saying, is there a signature there? Okay. We're in a hurry. Fine. And they also found that there tended to be a lot of reuse and that, even in known instances where malware, malicious software had been signed by digital signatures, and in the case of 111 which were used to sign those 189 samples, only 27 had been revoked, while 84 remained valid. And of course we've talked a lot about how crippled the certificate revocation system is. It just, unfortunately, it's not revoked by default. It's revoked if you take the time to check.

Well, if they're not checking to see whether the signature is valid in the first place, they're certainly not going to go do an Internet query to pull up the CRL, the Certificate Revocation List, if the CA even bothers to manage revocation, which in this case many of them don't because, as we know, our operating systems are, like, trust everybody. Trust all of these CAs. Any of them are able to produce signatures or to hand out certificates which malware authors can then use to sign their code.

So unfortunately, what's happened is what started out to be sort of a good idea, let's require signatures in order to attest to and assert a higher level of integrity of the software, it's just become useless. They found that a large fraction, in this case 88.8%, of malware families rely on a single certificate, which suggests that abusive certificates are mostly controlled by the malware authors, rather than third parties. That is, no one bothers to go steal somebody else's cert because you can just ask someone, some CA - and they're not all reputable, but they still exist. Ask them for a cert, and they'll sell you one, and then you can just sign your malware.

And so then, until it's determined to be illegitimate, AV, no matter whether they test or not, I mean, if they do test the validity of the certificate, they'll say, oh, this is fine. It passes all of our checks. And what we now know is many of them don't even do that. They found that at least 34 current antivirus products fail to check the certificate's validity.

Leo: I didn't even know there were 34. I'm close to all of them, I would guess.

Steve: So three prominent antivirus products - nProtect, Tencent, and Paloalto - detected the original unsigned ransomware samples as malware, but considered eight out of 10 crafted samples, that is, these researchers took unsigned malware. Those three

products all said, okay, yes, we see them as malware. These guys, these researchers, then stuck on some random signature from somewhere else that was invalid for that code, and eight out of 10 of the samples they tested were then passed as benign. Kaspersky Labs, Microsoft, Trend Micro, Symantec, and Comodo also failed to detect some of the known malicious samples when they had invalid and expired certificates attached to their code. They didn't even check the date on the certificate.

So basically it looks like these companies are in such a hurry to not slow down their scan that they hoped nobody would notice that, if software is carrying a signature, even if it's expired, even if it's like that they got from something else and stuck it on, I mean, where the signature doesn't even work because it signed a different hash than the code that it was stuck onto. Lots of products don't care. They go, oh, it's signed. Well, okay. Or rather it has a signature because signing implies that the hash matches, which this doesn't.

So anyway, I'm super glad for the exposure that these guys created. And what this allows people to do is perform similar tests. In fact, they created a site to back up their research paper, SignedMalware.org, just all one word, SignedMalware.org, where they run through all of the malware that they found, the certificates that are stuck on them, who's issuing these certs. And, I mean, it's not the issuer's fault. I mean, many of them are VeriSign and legitimate certificate authorities. The problem is anybody can get one. And now what we're learning is that getting one sends a signal to the malware that don't look any further. Don't scan this code. Don't. Just like, oh, look, we have a signature. Fine, keep moving. Wow.

Leo: It's amazing. Just amazing.

Steve: Yeah.

Leo: Easily fixed. I presume all antiviruses will start paying attention.

Steve: Yes. It will, yes, when the light is shined on them, they don't want to be in the doghouse. So they'll have to check the signature. They'll check the date. And they'll have to do a much better job of checking.

Leo: Did they say which ones passed? Like there were some that did it right?

Steve: I didn't dig into the actual paper. There is a full-length research paper that they've put out. And, I mean, they are naming names. So I wouldn't be surprised if they...

Leo: When they say Microsoft, is that Security Essentials? Is that the Defender that's installed with Windows 10? Or is that some sort of separate - because that would be disturbing, that Windows...

Steve: That would be disturbing, yes.

Leo: We're telling people to rely on the Windows antivirus as sufficient.

Steve: Well, and I wouldn't know - yes. I wouldn't change my advice because it looks like all the competitors are doing the same thing.

Leo: Nobody else is better, yeah, yeah.

Steve: So it's like, ah, yeah, okay. So Robin Linus has a GitHub page and posted a nice piece of research. Well, a little bit disturbing. It turns out that, as a consequence of some clever design leveraging, it is possible for a site you visit to determine whether your browser is currently logged into any other of many possible sites.

So let me explain how this works because it's not too complicated. We've often talked about how the same-origin policy, the so-called SOP, same-origin policy strongly restricts which web content JavaScript which has been received from a website, restricts what web content JavaScript received from a website is able to access. Specifically, JavaScript is then, once it's running in your browser page, can only fetch more things or do stuff with the site it came from, thank god. Otherwise it could be, you know, you'd be doing - you would be loading code into your browser that itself could go do anything else it wanted to on the 'Net, and havoc would reign.

So that's rigorously enforced except for images. Images, being benign and useful, are specifically excluded from same-origin policy, which you're able to use for many purposes. In fact, I even use it with SQRL in the spoof-proofing system that SQRL uses when you come onto a page where you can log on with SQRL. The JavaScript on the page queries to see whether you've got SQRL running in your system, and it does so by asking for an image, which is cross-origin because that code is from the site you're logging into, and we're checking on your own system.

So, I mean, it's a very common practice and can be very useful to allow images to be cross-origin. In this case, though, it creates a problem. The login mechanisms of most major sites - and Leo, you should go to, oh, I have the link here at the top of the story, yeah, "socialmedia-leak." If you click that, it'll bring up an immediate display of all the sites you are currently logged into.

Leo: Oh, my god.

Steve: Uh-huh. So this is like a social media privacy leak.

Leo: Facebook, Twitter, Gmail, Reddit, YouTube, Blogger, Disqus.

Steve: Yes.

Leo: Wow.

Steve: So this GitHub.io domain, this page on that domain has just been able to determine all the sites that your browser currently has session cookies for.

Leo: And this would be worse if I weren't using Linux. Some of these other things would also be logged into, obviously.

Steve: Right.

Leo: Like Dropbox. Wow.

Steve: Yes.

Leo: Let me do it on my - I'm going to do it on my...

Steve: Yeah, uh-huh. So what happens is that the login mechanisms of most major sites check every incoming request for the presence of a logged-in session cookie because, as we know, when your browser carries cookies for a given domain, every query it makes, it hands those cookies back. And that's how we maintain session state and logged-in-ness with sites. So if you go to a site where you're not logged in, you typically, if you try to go to any page there, you get an intercept, and you are instead taken to the login page to log yourself in.

And so that means that the behavior of the site is different. If you are logged in, and you for example ask for an image, it just gives you the image because that's what your browser would be doing if it were asking for a bunch of images to populate the page. If you're not logged in, and you ask for an image, it says, "Oh, we don't know who you are," and bounces you over to the referral page. So if you combine JavaScript's ability by design, not a bug, by design to be cross-origin for images, just those two features allows you to do what this page is doing, which is JavaScript running on the page asks for an image almost all sites have. And that's the favicon.ico.

Leo: Yeah, my site has it, sure.

Steve: Well, everybody does because that's where you get your logos in the tabs and on the browser title and everywhere. So everybody wants their site's logo to show up on browser tabs. And so favicon.ico is a file in the root page of all servers. So this script asks for that favicon.ico. If it succeeds, JavaScript will trigger an on-success result, which means you're logged into that other site. And if it fails, it's because you got a redirect because you're not logged in, and JavaScript goes, okay, we didn't get a successful image load. Which, now, okay. So that's annoying; right? But ads are allowed to run JavaScript in order to do their ad rotations and what, I mean, we often talk about what a problem it is that third-party ads hosted on benign sites can get up to so much mischief.

Well, now an ad that is being hosted on a site you're visiting is able to probe your browser to determine what suite of other sites you're logged into. So this is the kind of thing which sites could prevent by looking for this behavior, but no one's bothering. And this guy is hoping that his shining a light on this is going to bring it to people's attention

and begin to get this fixed, which would certainly be nice because it's a little shocking when you go to that page, and it's like, oh, wow, this third-party site knows all the places I'm logged into currently. I mean, it doesn't tell them who you are. But it's a form of fingerprint.

Leo: That's really, yeah, it's a favicon. Of course, what else? That's the one thing everybody has.

Steve: Yup, yup, yup. That's very clever. So, okay. As we were saying at the top of the show, the press went wild over the last couple days over this MINIX 3, which is the latest edition of MINIX, which is a minimal Unix which has been around for quite a while. It's meant to be very robust and sort of self-healing. I love it, there's actually something known as the "resurrection process" running in it, where if something crashes that shouldn't, it resurrects it and brings it back. It's got a very small kernel. Most of the stuff runs out in user space. So it's also a sort of a - it's a microkernel architecture which hasn't been messed up by commercial interests.

Leo: People who are familiar with Linux will of course recognize it as the inspiration for Linux because Linus - it was originally written as a teaching kernel, teaching OS by Andrew Tanenbaum.

Steve: Right. It was Tanenbaum, right, yes.

Leo: And it was what Linus, who was a com-sci Ph.D. student, was forced to use. And he one long winter Finnish night decided, I'm going to write a better MINIX. And so Linux really stands for Linus's MINIX, of all things.

Steve: Yes.

Leo: So that's kind of interesting. Now, it's come a long way from the educational platform, I gather. It's really - now it's a commercial OS, sort of.

Steve: Yes, it is. It is a very solid, mature, usable - and because it's so robust, it's used in embedded applications. So not surprisingly, when Intel was wanting to increase the power and capabilities of their embedded IME, the Intel Management Engine, they decided to put a processor running MINIX in the so-called PCH chip, the Platform Controller Hub. And in fact in the next page of the show notes I show a picture of the Platform Controller Hub. The press all got this wrong, of course. They were saying that it was the Intel processor that had this extra OS embedded in it. And it's like, no, it's not. It's part of the chipset support system that has this in it. And so, I mean, you're stuck with it. It's there. Well, you're kind of stuck with it. We'll talk about killing it off in two stories from now.

But I just sort of wanted to clarify, for anyone who caught this and was wondering what was going on, that in fact it's the OS, which we do have no access to. It is very mature. It's very small. The microkernel is like 600K. And so it would make sense that they would want something toned down. But when you hear things like the IME has a full networking

stack, it's like, what? In hardware? Well, no, it's actually MINIX running a driver for the NIC, the Network Interface Controller on the motherboard.

Leo: Here's a weird coincidence. Just as you were starting to talk about this, I got an update for my Intel Management Engine 11.8 firmware on my Windows machine. Wow. And it says, by the way, we're going to reboot you, so save your work. Ugh. This isn't related to this, I'm sure.

Steve: Well, Intel's not happy...

Leo: Oh, maybe it is.

Steve: ...that people are now beginning to disable it because there is, in the classic - and I love this - Internet collaboration that we have as a consequence of open source stuff, a guy named Nicola Corna has on GitHub something he calls the "ME Cleaner," the Management Engine Cleaner. And it's a beautiful piece of work. If you give this open source code that he's been working on and honing for some time, you give it a firmware image of the firmware from the BIOS, you give it your motherboard's BIOS firmware image, it will go in and remove the management engine.

Leo: That doesn't sound good necessarily. What are the consequences of this?

Steve: Yes. So there are two modules that have to remain, and one of them is the, not surprisingly, the boot module that gets things going. The other one is like the power-on management that gets clocks set and so forth so that anything is able to happen. But then there's all this other junk in there, which is the stuff that, like, grabs your network interface adapter and, as we've been discussing, has a lot of people worried from a security standpoint because there have been bugs in it. Remember there was a huge scramble because a serious security vulnerability was found four or five months ago.

Leo: Now I'm really worried. Okay. So I'm looking at this firmware update. There's, okay, it says "Before continuing please ensure the following. Make sure the AC adapter is firmly connected to system outlet. Make sure that a charge battery pack" - there's either a typo, or this person doesn't speak English very well. This is supposed to come from Intel. Now what am I supposed to do?

Steve: Yeah, yeah. And, boy, if that were malicious...

Leo: Oh, Mother McCray. I'm just glad you're here. I'm going to cancel. I don't think I want this.

Steve: Yeah. Well, I mean, what's it going to do for you? Probably, I mean, maybe you could do some research, verify that...

Leo: There were bugs earlier; right? I mean...

Steve: Correct, yes, correct. Okay. So as we know, nearly all systems allow the system's BIOS to be updated using software only. Many times when you get a motherboard it's like, oh, check to make sure you're running the latest BIOS. The common wisdom had been, and we talked about this at the time, don't mess with your BIOS unless you need to. But when there's something that you know a BIOS update fixes that's a problem for you, then, yes, apply it.

Okay. So the software-only fixing is regarded or called "internal flashing." But on most PCs, only an unprotected area of the flash file system, which excludes the management engine area, can be overwritten by software. Makes sense. They would not want you to be flashing their proprietary stuff. So working off of Nicola Corna's work on ME Cleaner, another individual has come up with a complete "Disabling the Intel Management Engine" guide, step by step. And Leo, you really should click this link and scroll through it because, I mean, if you thought you were, I mean, it really is a hardware hack. So what has to be done in order to do so-called "external flashing," in order to overwrite the Management Engine, is to literally reprogram the flash chip on your motherboard. And so he sets up a Raspberry Pi 3, Model B, as an in-system flash programmer. It reads the original firmware from the little...

Leo: Oh, my god.

Steve: It's a little tiny eight-pin chip sitting on the motherboard.

Leo: I am not doing this.

Steve: No, no. This is not for the faint of heart. It reads the firmware, extracting it from the chip into a file. You then give it to Nicola's ME Cleaner. Oh, and if you look, if you scroll down further, you could see the ME Cleaner running. Or I think you have to go to the ME Cleaner page, GitHub.com/corna/me_cleaner. I've got that link below that first one in the show notes. And you can see what ME Cleaner is doing as you run it, where it's recognizing, it's interpreting the firmware that you've given it, and it's unscrambling it and figuring things out and then blasting crap out of the firmware, saying, okay, we don't need that, and we don't need that, and we don't need that, and we don't need that. And we've got to keep this and this and this. Oh, and it also flips - remember that bit that the NSA required for their motherboards? There's a special bit that can be flipped which allows the BIOS to turn off IME. It also flips that, just so then you get a new BIOS setting to turn it off just for sort of belt-and-suspenders reasons.

But anyway, so this how-to extract - and there's the chip on the screen. You're showing it right now, a little surface-mount eight-pin chip. You get a clip, which you snap over its back in order to connect to the wires. And then if you scroll down a little bit further, you'll see the diagram of the chip showing power and ground and where the signals in and out are. And again, you need to be a hardware guy or be comfortable with hardware stuff in order to do this.

Leo: Or buy the Purism Librem laptop, which has this done already to it.

Steve: Yes. Well, and so my thinking is where this would be practical - first of all, if you were just a hacker, and you had an older motherboard that you wanted to screw around with, this would be fun. Or if you were an enterprise, where you had a thousand identical systems because you bought them all at once - yup, you've got the Pomona clip there, which is used for hooking to the chip.

Leo: I don't know why Burke has this, but we got it, yeah, yeah, yeah.

Steve: So if you had like a thousand of these identical systems and could spare, like risk destroying one because you might destroy it, but the benefit was, if you really cared about shutting down this unneeded Management Engine, if in your organization in fact it was not needed, if you weren't using it for its purpose, then you could try this on one motherboard, verify that it works, and then apply it to all of the rest of the many hundreds or however many motherboards your corporation has. So my point is that by amortizing the risk across the benefit of hundreds of motherboards, maybe it would make sense.

So anyway, what this does is it hardware reads the firmware into a file. You then hand it over to Nicola's ME Cleaner, which flips some bits and overwrites and removes a bunch of the junk you don't want. And then you, using the Raspberry Pi, rewrite the new firmware back onto the motherboard chip. Then you plug the power in and hold your breath. And if it boots, you no longer have IME running on that motherboard. Which is pretty cool, I think. Oh, and if it doesn't, then presumably you can reflash the original file and at least restore this thing to normal operation, and it wouldn't be any the worse for wear. But don't try this at home, kids.

Leo: Yeah, yeah.

Steve: Definitely for a high-end hacker, but also cool.

Leo: I doubt many will do this. You'd have to be extremely paranoid.

Steve: Yeah. So...

Leo: Continuing on.

Steve: So spoofing we've been talking about a lot lately. It's the unsolvable problem because it's probably the ultimate abuse of user trust of the world, or I don't want to put too much responsibility on the user and say "user inattention." I mean, we're all human. We're all inattentive. Sometimes we're in a hurry. Sometimes we don't remember to go check the URL and make sure of the domain we're looking at, or we don't hover the mouse over the link. The point is that too much responsibility falls to us, and that gets abused.

So there was again a lot of coverage over the last week about an extremely convincing WhatsApp fake which was downloaded more than a million times from the Google Play Store. And what's disturbing is that it was such a simple thing to do. Whoever this was who perpetrated it simply added a unicode space character to the end of WhatsApp, Inc. So that didn't show, of course, on the screen. And users said, oh, this is from WhatsApp, Inc. It says so right here. And a million people downloaded this thing.

Now, the good news is it wasn't, didn't do horrible things to them. It was simply - it required minimal permissions, and it was a whole bundle of ads for other apps. So people quickly learned that, okay, this isn't what I thought it was going to be.

Leo: How did this get by Google? That's my question.

Steve: I know.

Leo: They'd only have to run it to realize that; right?

Steve: Exactly. Exactly.

Leo: Terrible.

Steve: And so that also suggests that there's minimal curation going on. And when you consider the size of the store, how much stuff there is, it's almost understandable. Again, what they'll have to do is add some code to prevent trailing spaces or leading spaces or embedded unicode, you know, put some filters on that in order to try to reduce the spoofing. But this is just, I mean, this wasn't a huge security issue. But it could have been. And it just demonstrates that most users are, even if you checked to make sure that it was legitimate, as far as you could tell, that it was WhatsApp, Inc., which is what you'd expect. It's like, oh, okay, good, it's from WhatsApp. But, whoops, it's not.

Leo: Very disappointing.

Steve: It is, yes. An Italian researcher discovered that the macOS and Linux flavors of the Firefox-descended privacy-enhancing browser used by Tor was leaking their users' IP address. Whoops. Because of course the reason you use Tor is to bounce your traffic among onion routers and not reveal your true source, your true IP address. He named it TorMoil and released it privately to the Tor Project so that they were able to fix it before it became public. He's the CEO of a security firm named We Are Segment. And 12 days ago, on Thursday, October 26th, he let the Tor developers know, and they rolled out an emergency update version 7.0.8. Windows users were never affected, so only people on macOS and Linux.

And then the following day they posted, that is, the Tor Project posted: "The fix we deployed is just a workaround stopping the leak." Because they were horrified. So basically they just shut down the class of URL which was the file://. As people may know, we're used to http:// or mail://. It turns out file:// can be used to retrieve a local file on

your system. And a small glitch in the Tor privacy-enhanced browser for Mac and Linux allowed there to be a way to get the user's IP from using a file:// URL.

So what they wrote was, "As a result of that," that is, of the quick fix, "navigating to file://URLs in the browser might not work as expected anymore. In particular, entering file:// URLs in the URL bar and clicking on resulting links is broken. Opening those in a new tab or new window does not work, either. A workaround for those issues is dragging the link into the URL bar or on a tab instead." So I'm just guessing, but that sounds like something about the referrer because, if you were to click on a URL on a page that came from a file, maybe the recipient of that query in the referrer there was the IP; whereas dragging the link into the URL probably looks like a fresh launch rather than a referred URL. So that would make sense.

It sounds like what they did was they just quickly shut it down completely, just like broke it in order to produce, in the future, roll out a better fix. They just wanted to immediately nip that in the bud. And it's not a feature that would probably upset or be missed by many people. You'd have to be clicking on a link to a malicious site that knew to collect that information which was contained on a file on your system, so a rather circuitous means to get someone to do that. But again, they want to keep their Tor browser as watertight as possible and not allow that kind of leakage. So anyway, I thought that was interesting.

And at the same time we are about to have a substantial upgrade to Tor after four years of work. They're calling it the "Next Gen Onion Services," also known as Tor v3. It's now in alpha, has been for a couple weeks, and they're getting it ready for primetime. Within the Tor community this has been known as Proposal 224, and it does a bunch of things. It offers better crypto. They are replacing the use of SHA-1 and Diffie-Hellman and RSA-1024 with our favorite stuff - SHA-3, and they're replacing Diffie-Hellman with the Edwards 25519 Curve which SQRL of course also uses, and they're replacing RSA-1024 with Curve25519, both of which I had chosen a couple years ago when the whole concept of SQRL occurred to me.

So they're updating Tor's crypto to what's state of the art. They've improved the directory protocol, which as a consequence will leak much less information to directory servers than Tor has been, and we've talked about that modest leakage sometime in the past. Also another improvement will create a much smaller surface for targeted attacks on the directory protocol.

They've got a better onion address security to protect against impersonation, and in fact they've made it much longer. I have in the show notes an example of an earlier, the original onion address, which is what, [counts to 10], looks like maybe about 30 characters of address, or maybe 21 or something. The new one is 56. So it's much longer. In fact, that's the quick way for anyone to determine whether they're looking at a v3 onion protocol or a pre-v3, as all pre is a short link, and the new onion links are conspicuously longer, which is in order to create additional security and privacy. They've also developed a more extensive introduction/rendezvous protocol and really done a big revamp of the code. It's a much cleaner and more modular code base.

So this is basically four years of work and a significant bump to the Tor protocol, which is now, as I said, in alpha, and which we should be seeing coming out of beta before long, which will be great. And I forgot to mute some of my iDevices around me, so we're getting some background noise.

Oh, and as I mentioned at the top of the show, Signal.org, the Signal people behind the Signal super-secure messaging platform, have announced standalone Signal clients for

our desktops. If you go to Signal.org/download, you will see that page now boasting Signal for Android, for iPhone, for macOS, for Windows, and for Debian-based Linux. And they are formally deprecating the Signal for the Chrome browser. So that's been removed. The new desktop version of Signal runs independently of any browser, so that Firefox and Safari users will no longer need to use Chrome just to send and receive Signal messages, like using Chrome only as a client platform for the Signal client. And Chrome users will no longer need to have Chrome open just to be able to use Signal. 64-bit versions of Windows only are supported, from Windows 7, 8, 8.1, and 10; macOS from 10.9 and above; and Linux distributions supporting APT like Ubuntu and Debian.

So this is great. And it'll be interesting to see what, if any, pressure this puts on Apple to extend the reach of iMessage. Maybe they just won't care. I mean, in the same way they don't care about Flash, they'll just say no, we're not doing that. But as a Windows user, as I mentioned, I chafe a bit that I cannot have iMessage on my Windows desktop because I use it to glue a lot of my dialogues with my friends together.

And being able to link my friends together across all our desktops using Signal, which is as we know really, really, really well done - we talked about the Ratchet protocol some months ago. And I remember when I took a serious look at it, and we did a podcast on it, my first impression, as I was first looking at the beginning of the whitepaper, was that it was ridiculously over-engineered. I was reading, you know, this, it does this and this and this and this. And I was thinking, what the hell? Why? But by the time I got to the end, and I saw why they had done what they had done, it all pulled together, and I was very impressed and told our podcast listeners at the time. It's like, okay, these guys really did this right.

So we already know Signal on our mobile devices is arguably the strongest messaging platform available. Now we have it running and available on our desktops. And I would say, if you want to have a private ongoing dialogue from desktop to desktop somewhere, don't use Skype. Don't use anything other than Signal because it's open source, it's open protocol, it's been vetted like crazy, it was designed by very mature crypto people. It's the one to choose. And now you can use it on our OS desktops. So bravo to those guys for adding that to it.

And Leo, you were talking, after this came to light, about this Google Docs glitch which surprised some people by locking them out of their own private documents, which occurred last Tuesday. So it must have been you were talking with Stacey and Jeff last Wednesday, and I must have had the podcast playing while I was working on SQRL in the background because when I'm not writing English I'm able to listen to English in the background. So what happened was that some Google Docs users received an alert saying: "This item has been flagged as inappropriate and can no longer be shared." Others saw: "You cannot access this item because it is in violation of our terms of service." When, like, private people were just wanting to access their own docs.

This occurred, because it was on Tuesday, that was of course Halloween last week. Rachel Bale, for example, who is a high-profile individual, she's a reporter for National Geographic's Wildlife Watch, and she reports on wildlife crime. She tweeted the question: "Has anyone had Google Docs lock you out of a doc before? My draft of a story about wildlife crime was just frozen for violating their ToS," their terms of service. And unfortunately Google was a little - was deflecting a bit. They said: "Google's automated systems periodically scan certain file types in Google Drive that are shared with other users to detect abuse and protect users."

Leo: You could see why they would do that. It's almost, I mean, it's malware protection.

Steve: Yes, yes.

Leo: And it's only when it's been shared. It's not just your stuff sitting on your drive. They have to share it. You have to share it.

Steve: Okay. So probably she, for example...

Leo: She shared a link for collaboration.

Steve: Or maybe she shared the folder. And so, for example, you and I share the Security Now! folder so you can always just grab whatever that happens to be in there.

Leo: It kind of makes sense. I mean, on the one hand, of course, it's disturbing. But if somebody is sending out share links, there's the potential that those share links could contain malware.

Steve: Yup.

Leo: So I can understand why they might want to scan those.

Steve: And so what was the glitch? How did this, like, I mean, because it was a [crosstalk] mistake.

Leo: The glitch was that they - yeah. So what Google said is we scan - antivirus scanning, malware, phishing detection. But she just said something, I don't know what happened. It may have been, well, the glitch might have been that it was a false positive; right?

Steve: So, yeah. So it may have been - so maybe their heuristics were updated. I heard something about something, a code push that caused the problem.

Leo: Right. So that's probably what it was. I mean...

Steve: And so this has always been going on, but people weren't, because it wasn't false positiving, people were unaware that that could happen to them.

Leo: Yeah. So, I mean, we were concerned about it. Obviously you don't want to know that what you have presumed your Google Docs to be private, you don't want them not to be private. But this is, upon further digging, it's only when it's being shared.

Steve: Right.

Leo: And so that kind of makes sense. That's to protect people from sharing malicious docs.

Steve: Right. So under the topic of why we can't have nice things, or in this case secure things, US-CERT has added a vulnerability and some mitigation after discovering that, believe it or not, an audio driver manufacturer - there's a company Savitech, S-A-V-I-T-E-C-H, which provides USB audio drivers for a number of specialized audio products. When you install their driver into your system, some versions of this driver package silently install their own SaviAudio root CA certificate into your Windows trusted root certificate store. So it's like, you know, okay, wait. What that says is they were unwilling to buy a certificate from a CA that's already trusted.

Leo: Oh, geez.

Steve: So instead they basically did a self-signed cert. They said, no, we don't want to - we're not going to buy a certificate from a CA. We're going to take advantage of the fact that the user will click Okay no matter what we show them, and up comes UAC. Do you want this driver to install stuff into your machine? What are you going to say? You say yeah, I need this audio driver. Well, along for the ride is their own root CA, which now means that anything they choose to sign will henceforth, forever probably, be trusted by your computer, just as if it was a malicious CA. And to make matters worse, it turns out this was done for Windows XP, which is no longer necessary, but they never bothered to remove it from the installer on later operating systems.

Leo: I guarantee you somebody got a raise for thinking of this. Oh, boss, here's an idea.

Steve: Yup. We can save 100 bucks.

Leo: Yeah, we'll save money.

Steve: Wow. Wow.

Leo: That's so funny.

Steve: So, okay. As I said at the top of the show, my number one wish list item for this

Christmas would be that we solve the problem of updating IoT things, the Internet of Things devices. What just happened, I think it was October 20th, or I'm sorry, October 30th, I had that date in my head. And so just two weeks ago, or actually Monday before last, was a draft from a team of three at ARM, which is where you want it to be, the ARM guys know what they're doing, a lot of these IoT things are ARM-based. It's a proposal, it's a draft, Internet RFC working draft, to propose a set of standards for IoT updating mechanisms. I read the whole thing. It is well thought through. And I'm just, I mean, all my fingers and toes are crossed. It would be an incredibly welcome addition to this entirely vacant need that we have at the moment.

I'm not going to go through it in detail because at this point it's premature. But if it ever happens, if it gets assigned an RFC and becomes any sort of standard, we will absolutely be talking about it. But I'll just read just a few words here from the introduction. They said: "When developing IoT devices, one of the most difficult problems to solve is how to update the firmware on the device. Once the device is deployed, firmware updates play a critical part in its lifetime, particularly when devices have a long lifetime, are deployed in remote or inaccessible areas, or where manual intervention is cost prohibitive or otherwise difficult."

So they say fixes to bugs in software that can be applied to the device with a firmware update are needed. New functionality can be added to the device with a firmware update. And whatever this updating process turns out to be, it must ensure that the firmware is authenticated, attempts to flash a malicious firmware are prevented, and that the firmware can be confidentially protected, that is, the firmware's confidentiality can be protected so that attempts by an adversary to recover the plaintext binary can be prevented.

So they go through in this proposal the things that would be needed. For example, there needs to be side-by-side firmware so that there's the current boot firmware and a second sort of sidecar receptacle where an entire next iteration can be downloaded and installed, I mean, like be completely ready. That way if there's a power outage during the firmware update, basically, Leo, the things your laptop was just warning you you'd better - it's got to be plugged in. You've got to have a battery, blah blah blah.

These guys are saying, okay, we don't want those to be limitations. They may be practical for a laptop. They're not practical for stuff that may be autonomously updating itself when you pull the plug because you want to move it to a different room. You don't want that to brick your device. So the idea would be that a shadow firmware region would exist that would receive updated firmware. Its signature would be verified. It would be completely confirmed. And then hardware would reboot the device, and a watchdog would verify that the device came up under the new firmware. And if it didn't, the device would restart again with falling back to the old.

So, I mean, my point is they nailed this. They sat down, they said, okay, how do we create a robust architecture which is still inexpensive, so that it's actually going to be adopted, but which incorporates everything that we need in order to set this as a standard. So I'm holding my breath because it would be great if we solved this problem. And then, if it were a standard, it would be something that consumers could look for in the same way that the Wi-Fi Alliance has their logo and Bluetooth has their logo, give this thing some sort of a logo so that people can verify that this has self-updating firmware technology built in. And, boy, that would be great.

Leo: Little Stevie G, we call him.

Steve: Not what we're calling him. So two contrary opinions about Windows 10 Controlled Folder Access that we discussed last week as a feature coming to the Fall Creators Update of Win10.

Leo: Oh, interesting.

Steve: First of all, Jakob Engblom, he wrote and sent me a tweet saying: "Windows 10 CFA is off by default for a good reason. It breaks a bit too much stuff." And then he sent his analysis, a link to his analysis.

Leo: It's common with a lot of these things like ASLR and stuff like that. It's just stuff breaks.

Steve: Yes. And so I think that's why it makes sense, as we said last week. This is what Microsoft generally does is they'll put it in there. They'll have it off by default so that the user takes some responsibility for turning it on and hopefully then knows what it is they did when stuff doesn't work anymore. Whereas - I liked his handle. His name is probably Barry Wallis, so his Twitter name is Cranbarry Wallis. Anyway, he said: "@SGgrc, I turned on Controlled Folder Access. There were two old Nikon photo editing programs I had to whitelist, but that was it. It works great."

Leo: Interesting.

Steve: So my takeaway is...

Leo: Try it.

Steve: If you're concerned - yes, try it. If you're concerned about cryptomalware getting into your machine and wreaking havoc, boy, I think it makes sense to have your stuff that's most vulnerable protected by a whitelisting system.

Leo: Of course it's what Apple's done with iOS since day one.

Steve: Yes, yes.

Leo: Google tried to add it to Android. It broke a bunch of stuff. They backed off a little.

Steve: Because it's difficult to retrofit. That's the problem is adding it later.

Leo: Yeah, yeah.

Steve: Yes. And so a couple little bits of miscellany. There's a great blogger and tweeter who tweets from @SwiftOnSecurity. And I just love this. Someone forwarded it to me. So his tweet is: "We were the victim of a very simple attack. It was through management's lack of focus on security that this happened," dot dot dot, said no company ever.

Leo: You must follow @SwiftOnSecurity; right? She's great.

Steve: Of course.

Leo: And by the way, it's not a he, it's Taylor Swift, the famous rock star who has a sideline doing security. Many people don't know that.

Steve: Yup. I just got a kick out of that. "We were the victim of a very simple attack. It was through management's lack of focus on security that this happened."

Leo: You'd have to be nuts to [crosstalk].

Steve: Said no company ever. No.

Leo: It's true, though, almost every time. But okay.

Steve: Yes, exactly. Yes, exactly, that's what we keep seeing. Chris Duncan, who is a listener, made an entry, he tweeted to me, on Twit's wiki site with a curricula of Security Now! episodes in our various series.

Leo: This is so nice.

Steve: It is so cool.

Leo: Now, it's in the Talk section, unfortunately.

Steve: Correct.

Leo: And I guess that's - maybe he doesn't have editing permissions or something? I don't know.

Steve: Probably. Maybe we could link to it in the notes for this...

Leo: Well, I will move it over into Security Now!, is what I'll do. It's fabulous.

Steve: So what he's done is he's broken down - oh, it is, it's just great. He's broken down the podcast history into topic groups. He's got VPNs, that is, all of the podcasts where that was the major topic. Internet and LAN Technologies is another. Cryptography is another. Virtual Machines and Sandboxes he's separated out. Web Code Injection as a topic. Designing a Computer, which is the one that we had talked about before. And then I think four on SQRL. So thank you, Chris. That's certainly useful, and I wanted to share it with all of our listeners, people who have joined us later thinking, well, okay, I know there's a lot there, but we're on Episode 636, so could you give me a pointer? Well, now we've got pointers.

Leo: So go to wiki.twit.tv. This is a media wiki we've maintained practically since the very beginning. And right on the front page we break it down into shows. Of course Security Now! is there. But Chris listened to our conversation from last week, and I think this is exactly right. I don't think it's on the front page. No, you're going to have to go to the Talk section. But you can easily do that. If you go to the page and click Discussion, you'll get the discussion and then scroll down. And I don't know, maybe he proposed it for discussion. I mean, I don't want to move it over if we can't keep it up to date. So basically the wikis are community run.

Steve: Ah, right.

Leo: But this is really, really great. So wiki.twit.tv.

Steve: Okay. And Leo, the best CYA bit of jargon I think maybe I've ever heard is what we now call "differential aging."

Leo: I know. Oh, man.

Steve: Oh, my lord, isn't that wonderful?

Leo: That's Google's circumlocution for OLED burn-in.

Steve: Yeah. So we all know that once upon a time, well, many of us know, the old-timers, those whose hair is not quite as dark as it used to be, that phosphor on CRTs would age. And in the old days, the original reason for a screen saver, the reason it was called a "screen saver," was it was saving your screen from having a static image burned into it. And what would often happen is, if it was a kiosk or an ATM or just if you generally had things that never moved on your screen, so that the same thing was always being there, the phosphor would age, and over time as it aged it became duller. So, that is, dimmer.

So then, if you put up something white or something different, you could see a ghost of what was normally there because our eyes are so sensitive to that. Against a white background you could see windows that you used to have or the scroll ticker along the bottom of the screen or whatever. So that's what screen savers were saving was they were literally saving your screen because after 10 minutes, rather than letting it just sit there burning, your computer would switch it to something moving around in order to

also give you some privacy, but mostly to prevent your screen from burning.

So what now we're seeing, now that we have a new technology which is not an optical shutter, which an LCD is, now we have something which is aging, which it turns out, if you don't take proactive measures, will dim over time, that is, OLEDs will do that. We have a new screen burn-in problem that's come back. And technically, yes, it's differential aging, meaning that pixels that are more illuminated age differentially to those that don't. But anyway, I got the biggest kick out of hearing you guys talk about it, I think it was over the weekend, and then Allen Butler...

Leo: Crazy.

Steve: ...said, "I've seen the screen differential aging on my original Google Pixel, and a friend has it on his original Pixel XL, as well."

Leo: Yeah. All these OLED screens have it as a potential problem. Google's pushing out an update right now that will, they say, fix this in a number of different ways. So we'll see.

Steve: And as I understand it, it just does a one-pixel shift? It just sort of blurs it a little bit?

Leo: Well, that's what I was told Samsung does with their phones. Pixel's going to do other things, including dimming the dock when it's not in use. So, I mean, that will help. So most people still don't have their Pixel 2 XLs. I don't have mine yet. The first batch went out. But I think this is...

Steve: And what do we know about the X, the iPhone X?

Leo: Apple says on their help page that it is, you know, all OLEDs are susceptible to burning, so they recommend you set the screen off time to the lowest possible, 30 seconds.

Steve: Interesting.

Leo: They recommend you - screen off time. What was the other thing they suggested?

Steve: Not leave sitting on the home page where it's always going to be the same stuff?

Leo: Yeah. What you want to do is really turn off the screen whenever possible.

Steve: Yup.

Leo: But that's for battery life, too.

Steve: Well, yeah, for battery life, yes.

Leo: There was one other thing they suggested. Oh, yeah, don't turn up the brightness all the way.

Steve: Ah, right.

Leo: For sure, most importantly, make sure you've turned on auto dimming, the ambient dimming.

Steve: Yes.

Leo: The problem with OLED is the brighter it is, the better it looks. You really - it pops.

Steve: It's gorgeous, isn't it.

Leo: Yeah. So I, you know, I hesitate to turn it down too much. We'll see. Yeah, I don't think it's going to have the problem. I mean, I've never had burn-in on - noticed, anyway. See, this is the point. Having it versus noticing it. I've never noticed it on a Samsung phone. I've had OLED screens forever.

Steve: No, Leo, Apple invented them.

Leo: Well, it's funny. This is a Super AMOLED screen. This is a Samsung screen. Same kind of screen, although to Apple's specs, that Samsung uses on its S8 and Note 8. I don't know. I don't know. My suspicion is you won't have a problem. I have an OLED TV. I've never gotten any burn-in. Or at least not, again, not that I noticed.

Steve: Yes, yes.

Leo: And part of the problem with this is people are really looking for it. Yeah, we'll see. I'm getting a Pixel 2 XL in about a week, so I'll let you know.

Steve: Cool. So I got a nice note from a Yann Fitzmorris. Actually, it was forwarded to me by Sue, my sales gal. He sent it to her under the subject "Another success story: SpinRite data recovery." This one is kind of heartwarming. He said: "Dear Steve and GRC team, I purchased SpinRite a few years ago and have been using it to keep my drives in good health. I personally have never had to use it for data recovery." And actually we'll

find out in a minute because apparently he has a dedicated SpinRite computer that he runs his drives on.

He said: "I personally have never had to use it for data recovery. However, a friend asked for my help this week because her laptop would no longer boot to the login screen. Her laptop contained the only copy of pictures and videos of the first two years of her daughter's life. It wasn't looking good for the patient," he wrote. "When I plugged the drive into an external dock, no OS would recognize the drive. I used my dedicated PC for SpinRite" - I guess that's what he means, dedicated PC for SpinRite - "plugged in the drive and ran Level 2. Success. We plugged the drive into the dock, and we were able to recover all pictures and videos, over 70GB. Needless to say, my friend will now seriously consider a backup solution, and I'm hoping she will buy her own copy of SpinRite to show her gratitude for this amazing product. Thanks again for all your hard work and research. Love the Security Now! podcast, as well. I've been a listener since 2015. Regards, Yann Fitzmorris." And, Yann, thank you for sharing your success. Always appreciated.

Okay. ROCA Pain. I guess we should have expected this. And of course I'm put in mind of something that - this is a phrase I hope Bruce invented. I love it. We credit him with it always, and he deserves it. This is Bruce Schneier, of course, where he said: "Attacks only ever get better. They never get worse." And when I was writing this, I was thinking of Gibson's Corollary, which is something I've told people over the years: If the car you're driving ever starts making a different sound, it's probably not an improvement. That is, cars don't get better, they generally get worse. And so is the case for attacks on crypto.

So three weeks ago the knowledge of this ROCA attack went public. And what we know is that the researchers who forensically examined a whole bunch of public keys that were being generated by this Infineon crypto library discovered to their credit that these were not robust private keys, that it was possible to factor the private key - I'm sorry. These were not robust public keys; that it was possible to factor the public key down to its primes, which revealed the private key. Which is exactly what this technology, the RSA technology, was designed to prevent.

So it turns out there are other skilled cryptographers on the planet. Their original disclosure three weeks ago estimated, as we discussed at the time, that it would cost an attacker who was renting time on a commercial cloud service an average of, for the smaller keys, the 1024-bit keys, $38 and 25 minutes. Much harder for a state-of-the-art 2048-bit key, about nine days of compute time costing about $20,000. So at the time we said, okay, that's really not good. Remember that it's supposed to be way, way, way, way more than the age of the universe squared. So this is far short. Nine days? That's a lot quicker than the expected age of the universe.

So we also know that many organizations who are known to be using these keys vulnerable to ROCA have largely downplayed the severity of the weakness, claiming that it was complicated and not inexpensive. And also even the guys in Estonia said that large-scale vote fraud is not conceivable due to the considerable cost and computing power necessary for generating the private key, cracking the public key down back into its components again.

But turns out, as I said, there are other gifted cryptographers, and particularly Dan Bernstein and the woman he often works with, Tanja Lange. They came up with a much more potent attack. Dan, remember, is the originator of the Curve25519, so definitely someone who knows his crypto. They reported that they developed an attack that was 25% more efficient than the one created by the original ROCA researchers. Okay. And their new attack was solely the result of Bernstein and Lange reading the original

research paper, which at the time omitted specifics of the factorization attack in an attempt to increase the time that hackers would need to carry out real-world attacks. So they said, yes, we figured how to do this, but we're not going to tell you how.

Well, they didn't have to tell Bernstein and Lange how to do this. These guys figured it out themselves. So after creating their more efficient attack, Bernstein and Lange submitted it to the original researcher, saying, "Guys, we've got a better way to do this." And then upon receiving that, the original researchers since privately disclosed their own revised attack that's as much as four times more efficient than what they originally published in their paper.

So that suddenly drops the 2048-bit key from nine days to, what, two and a half, and from $20,000 to $5K, making, again, I mean, it was already within reach for anybody who cared to crack one. Now it's four times easier. And what this suggests is maybe we still haven't seen the result because the more you look at some of these problems, the easier it becomes to overcome them. So as a consequence of all this, on Friday Estonia's Police and Border Guard suspended the entire set.

Leo: Oh, that's too bad.

Steve: 760,000 ID cards.

Leo: Can they fix it?

Steve: Well, no. They're now...

Leo: Because it's in a chip; right? It's kind of built-in.

Steve: Yes, exactly. It is in the hardware embedded in the chip. So they are now issuing cards which use elliptic curve cryptography instead of the vulnerable RSA keys.

Leo: Well, I'm glad I didn't get my Estonian ID card.

Steve: And remember that we've talked about how forward-looking Estonia is.

Leo: Oh, I was going to get one. I was in Estonia last year, and I meant to get one. I just didn't have time.

Steve: Yeah. They use it for voting, for border crossing, as their ID card. All kinds of things are tied into this. And they've had to say, whoops, we are just going to remove them all from service, three quarters, more than three quarters of a million cards. But to their credit, they're going to reissue them using elliptic curve crypto, which isn't vulnerable to any sort of a factorization attack because it doesn't use prime factors as the hard problem that needs to be solved in order to protect the key. So, yup, we could have anticipated that something like this would happen because, as Bruce said, "Attacks

only ever get better; they never get worse." And this podcast also only ever gets better.

Leo: That's a nice coda for the show. "It only ever gets better." Well, I am very happy to have, even though we started late, to have gotten everything in. It's not even 4:07 yet. We could have gone for another few hours. You sure you want to stop?

Steve: We're good.

Leo: All right. We are good, very good. Steve Gibson is at GRC.com. If you go there, you'll find this podcast, of course, audio plus transcripts, very nice transcripts that he pays for. Thank you, Steve. But you'll also find many, many other wonderful things. Of course SpinRite, the world's best hard drive maintenance and recovery utility. Got to have that if you have a hard drive. You've got to have SpinRite. Even an SSD you've got to have SpinRite.

Steve: Yup, it works there, too.

Leo: But SQRL's there, his perfect, what is it, better sleep formula? All the stuff that Steve does.

Steve: Yup, the Healthy Sleep Formula.

Leo: Healthy Sleep Formula, and Perfect Paper Passwords, Shoot the Messenger, DCOMbobulator, ShieldsUP! and on and on and on. Steve just has loaded that site with goodness. There's a feedback form there if you want to leave a message for Steve, GRC.com/feedback. But probably the best way now is using his Twitter account because he accepts DMs from strangers, bold man that he is.

Steve: Well, and there's a lot of nice dialogue that goes on in public, too.

Leo: No, I think it's great. @SGgrc. @SG, that's his initials; and GRC is the name of the company, GRC.com. @SGgrc. Tweet him there. Ask him questions. DM him if you've got a tip. And those questions make it in, as you can see, into the show at some point. We also have, of course, copies of the show at our site, TWiT.tv/sn for Security Now!. And you can always subscribe to your favorite podcatch client, and that will get every episode, and that way you can start building your collection. Collect all 636.

Steve: If you dare.

Leo: Each is a unique snowflake. If you want to watch us do it live, you're more than welcome to. There's a couple ways you can do that. You can join us in the studio,

and we had some nice visitors in the studio today, and we appreciate that. Just email tickets@twit.tv so we can put a chair out for you and so that our security guard doesn't shoot you on sight. Always good, you know, that's always a good thing. No, he won't do that. He's very nice. But do email us, tickets@twit.tv. The studio is not always open and accessible. You can always watch live on the stream. That's always safe. TWiT.tv/live. We're a little jumpy around here after all the gunplay that's been breaking out all over the country. We try to keep this place safe. I don't want anything to happen to my employees.

Steve: Well, and I remember that the TWiT Studios in South San, I mean, security is always present when I've seen TV studios.

Leo: Yeah. Yeah, TV studios have to do it. I thought we didn't have to as a podcast, and then somebody swatted us, and I thought, yeah, I'm going to have - I don't want to tell you the story on the air, but it was nasty. So ever since, it's been more than a couple years now we've had - Moe is a great guy. Marine, former Marine. Very good with weapons, small arms.

Steve: Don't mess with Moe.

Leo: Don't mess with Moe, that's all I'm saying. I don't want to scare people. We do more than welcome you; right? But, you know, I just want you to know, I do want to scare bad guys. Let's put it that way. Those people I want to scare.

Steve: If you're a bad guy, don't bother.

Leo: It's a shame, isn't it, we have to do that. But we do. I just - it's not that I feel fear for myself. I just don't want my employees to feel unsafe. So you can watch without coming to the studio. The stream is TWiT.tv/live. If you do that, join us in the chatroom, irc.twit.tv. To date, no shots have been fired there. You can also - what else? I guess that's it. I told you how to download it. There's nothing else you can also do. Are we doing - have you thought about our holiday show, what we're going to do for that? We can do a best-of. Should we? You don't want to work the week after Christmas.

Steve: I don't. But I do have some videos of our earlier...

Leo: In the past you've done some fun stuff, so that's why I wanted to give you the option.

Steve: Yeah. I'll try and find something fun.

Leo: We're putting together best-ofs of the other shows at TWiT.tv/bestof. But this

show, I don't think we've ever done a best-of.

Steve: We haven't, no. There's always been - yeah.

Leo: You shared a lecture last year that you gave, a speech?

Steve: Yup. That was...

Leo: And of course there's the Portable Dog Killer.

Steve: Yup, we've done that several times. I have some early TWiTs from South San Francisco, and that might be fun, just sort of as a blast from the past.

Leo: Really.

Steve: Oh, yeah.

Leo: Ah.

Steve: That was my deal was I would always get tapes from - literally VCR, you know, VHS tapes.

Leo: From The Screen Savers.

Steve: From The Screen Savers.

Leo: I think we could show those without getting in trouble with NBC Comcast Universal. Why not? It's always better to ask permission later; right?

Steve: Oh, and apologize. We're sorry.

Leo: It's better to apologize than ask permission. That's what it is.

Steve: That's right.

Leo: Steve, always a pleasure. Have a great week. We'll see you next time on Security Now!.

Steve: Okay, my friend. Thanks. Bye.