SECURITY NOW!

Transcript of Episode #610

Intel's Mismanagement Engine

Description: This week Steve and Leo discuss the long-expected remote vulnerability in Intel's super-secret motherboard Management Engine technology, exploitable open ports in Android apps, another IoT blows a suspect's timeline, newly discovered problems in the Ghostscript interpreter, yet another way for ISPs and others to see where we go, a new bad problem in the Edge browser, Chrome changes its certificate policy, an interesting new "vigilante botnet" is growing fast, a proposed solution to smartphone-distracted driving, ransomware as a service, Net Neutrality heads back to the chopping block (again), an intriguing new service from Cloudflare, and the ongoing Symantec certificate issuance controversy. Then some fun errata, miscellany, and some "closing the loop" feedback from our terrific listeners.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-610.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-610-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Lots to talk about, including a bad bug in Microsoft's Internet browser, a zero-day that's been exposed already, a mess-up in Intel's chip that's been a problem for almost a decade and is a big security flaw, and a lot more stuff like that. You know, it's a nightmare. But we'll talk about all of it and how to protect yourself, next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 610, recorded May 2nd, 2017: Intel's Mismanagement Engine.

It's time for Security Now!, the show where we cover your privacy and security online with this guy right here, Steve Gibson, the man in charge, GRC.com, and our security guru since 2005. Hi, Steve.

Steve Gibson: Hey, Leo. Great to be with you again for Episode 610.

Leo: Wow.

Steve: At the beginning of May. And in fact it was interesting that our main topic occurred on May 1st because this is truly Mayday for Intel.

Leo: Oh, boy.

Steve: The title of today's podcast is Intel's Mismanagement Engine. So we're going to discuss the long-expected remote vulnerability in Intel's super-secret motherboard management engine technology. Also there was a paper given at a European security conference, an IEEE conference, where five researchers from, I think it was the U. of Michigan, I've got it in my notes, took a look at something that no one had looked at closely before, and what they found was worrisome. And that's exploitable open ports in Android apps.

Leo: Oh.

Steve: We have another instance of an IoT device blowing a suspect's BS timeline. And you can just imagine, when you read through the details of this, they're just so screwy. I'm sure when the police came out to investigate they were, like, looking at each other, going, oh, come on, we're supposed to believe this? There are some newly discovered problems in the widespread Ghostscript interpreter. Yet another way for ISPs to see where we go online, another one that sort of escaped my overall summary, so I wanted to mention that. There's a new bad problem in the Edge browser which is very worrisome and was disclosed irresponsibly by an Argentinian researcher, for which there's no fix, and there are proof-of-concept exploits, that allows - well, we'll get to that. Bad.

Chrome is changing their certificate policy, which is interesting. There's a very large and suspiciously well-designed new botnet growing in size that's being called a "vigilante botnet." An interesting proposed solution to smartphone-distracted driving. The emergence of ransomware as a service. And of course we have to talk briefly at least about the concerning reversal by this new administration on the previous administration's position on Net Neutrality. And this is worth rallying all of our listeners and everyone within reach as we approach the time for public opinion later this month, after May 18th, because it's back on the chopping block again.

There's an intriguing new service from Cloudflare for protecting IoT devices that we need to talk about and of course remind our listeners that they are a sponsor of the TWiT network. There's of course then the ongoing controversy over the Symantec certificate misissuance and what the browsers should do to deal with it. And as if that wasn't enough, we have some fun errata, some miscellany, and then a little bit of "closing the loop" feedback from our terrific listeners. So, yes, 610 and going strong.

So I forgot to mention that we do have about a two-minute audio clip to play.

Leo: Uh-oh. Okay.

Steve: It's wonderful, following on from last week's very popular Turbo Encabulator. However, whereas that one was a bunch of mumbo jumbo gobbledygook, this is actually completely mathematically correct, but wonderfully obtuse. Anyway, it's missile guidance explained. And it's just audio, so there's no visual, so it works in the podcast.

There's another one that I just tweeted out that's more like the Turbo Encabulator that involves the Muppets' Cookie Monster, which is - unfortunately it's completely visual, but it is just hysterically funny. I had no idea the Cookie Monster could be so expressive. Anyway, but that's not what we're going to show. I put it in the show notes, and I tweeted it because - and you can also just google "Muppets Analytical Computer," and you'll find it on YouTube.

But the third link from the bottom of the second page, from the second-to-the-last page, the missile guidance explained, is just two minutes of audio that we will play when we get to it. But I wanted to give you a heads-up.

So our Picture of the Week has been in my queue. I have a backlog of Pictures of the Week, and nothing jumped out, so I pulled this one from the Security Now! backlog which I thought was interesting. And it's apropos a story that we'll be getting to about the open ports which it's noting that that - this shows a graph of four different OSes: Android, Windows, iOS, and OS X from March of 2012 through March of 2017, so essentially a month ago. So five years during which time the percentage of Windows OS drops from - it looks a little higher than 80, down to 37.91%.

But during the same time, Android moves up from looks like around 4% up to 37.93 - in other words, higher than Windows. More Android than Windows. And during this same time iOS sort of putters along, slowly growing from it looks like a little more than Android, but now way behind. It drifts. It maybe goes up from 5% up to around maybe 12 or 13. And OS X pretty much floats around below 10 and dropping down just a little bit. But anyway, just sort of interesting that, as this graph demonstrates, Android is now "the most popular," at least in terms of instances in the world. I'm not sure I would call that popularity, but in terms of count, since so many devices are Android-based. But that's significant. And of course it means that the security of Android, as we know, is important.

Okay, so this week's top story, I titled it "A True Mayday for Intel." About a year ago - I looked back through the transcripts backlog, and I couldn't find the specific podcast where we discussed this in depth because we've discussed it many times. We've touched on it many times. And that's the so-called Intel Management Engine, which exists in a number of different forms. There's something called IAM, which is Intel Active Management; SBT, which is Small Business Technology; and ISM, which is Intel Standard Manageability. It's been around since, like, 2008 with the - and I had it written down. I don't see it. Is it the Kalem? Or I can't remember which processor.

Leo: Nehalem?

Steve: Yes, that's the one.

Leo: Nehalem, yeah.

Steve: From them through Kaby Lake so, like, up through just now. And the ME has had different versions, this Intel Management Engine. But versions from V6 through 11.6, which is current, are the area of concern. The problem, as we discussed it before, is that this cannot be turned off. There's no way to disable it. You can't turn it off in the BIOS. You can enable additional features; but the baseline set of features, there's just no getting around it. It's a separate ARC processor that actually runs in one of the Intel chipset components that surrounds the main Intel processor that does all the memory management and I/O glue and slot management and so forth, providing USB and PCI functions and, you know, BIOS and all of that. It's always running.

Intel has gone to tremendous lengths to keep it a secret. So it's not open. It's never been subject to scrutiny. People have been for years worried about it and chipping away at it. But as we discussed previously at length, Intel did it, like used every trick in the book to hide what this is. And that alone is a concern, the idea that there is something in all motherboards for the last nine years, from 2008 on, which is outside of all non-Intel scrutiny. The protocol is not documented. It's available under NDA and has been licensed to, like, three companies. Unfortunately, one of them is Symantec. Let's hope they can do a better job with that than they have with certificate issuance.

So there are a few companies that provide enterprise access functionality that allow enterprises to manage their deployed machines throughout the enterprise. So not just servers, but laptops, desktops, tablets, anything essentially with an Intel chipset for nearly the past decade. And in fact our listeners will remember that I was pulling my hair out for a couple weeks. I was bringing up a new Intel-based 2U server about a year ago, and it was causing - over at the Level 3 datacenter. And I was getting these IP address conflicts. There was an ARP storm, and interfaces were fighting each other. And just nothing I could do. I couldn't turn it off. I couldn't figure out what was going on.

Finally, because these machines have multiple LAN interfaces, when I moved it from LAN 1 to LAN 2, that is, away from the primary NIC, all of that went away. And I remember, when we were discussing all this at the time, verifying that only the primary NIC on multi-NIC motherboards has this IME interface, and that what anyone could do would be to switch to a secondary or tertiary, anything but the primary NIC, and you would be okay. Because there's otherwise no way to turn this off.

Well, the other shoe has dropped, and we now have a confirmed exploit. I mean, this is what everybody was worried about, and this is as bad as it gets because unlike current versions of OSes themselves, and to an even greater degree browsers, and to some degree even apps, the motherboard BIOS, while there may be patches available, there isn't an auto-patching mechanism. And even though only enterprises really need or use this, this is the conundrum, is it's on and cannot be disabled through any means.

And while I was putting all this together, I dug into the background, thinking, you know, is there a scanner? Is there some way that we could check? The problem is, this is the problem with anything that is secret and proprietary and rigorously undocumented is nobody knows anything about this, except we now have a confirmed exploit. I found one site where some guy for years has been pounding on Intel, telling them this is a problem, and they've apparently just been ignoring him.

Leo: That's what's really frustrating is that this has been known.

Steve: Yes.

Leo: For years.

Steve: Yes.

Leo: That's really frustrating.

Steve: And Intel's just, oh, that's not a problem, that's not a problem. So suddenly it's Mayday. They have issued just yesterday, on May 1, 2017, patches for all of their firmware for all of these motherboards. And while that's the good news, the problem is they're all, I mean, like I've got, I just found, I just checked my Lenovo X1 Carbon. It's got it, and it's in there, and it's running. And I don't want it. But so it'll be one - as soon as I'm through with the podcast I will see about whether - and I'll talk about it next week, what I find. And I'll check GRC's Security Now! newsgroup because I'm sure that the people there will be interested in finding out whether there are firmware updates for their systems.

Leo: How could you patch this? Because isn't this in hardware? I mean...

Steve: No, it is, I mean, it's "deep firmware" is probably the way to describe it. So there are versions, and they do have patches. But there is no - and an enterprise could deploy these patches. But just so people understand, this is a rootkit. I mean, that's the best way to describe it. For my notes I wrote: "Recent Intel x86 processors implement a secret, powerful control mechanism that runs on a separate chip that no one is allowed to audit or examine. When these are eventually compromised, they'll expose all affected systems to nearly unkillable, undetectable rootkit attacks." And the guy I'm quoting said: "I've made it my mission to open up this system and make free, open replacements before it's too late." And that's what we - we talked about this last year. There is a project to replace this with a publicly available, open source solution.

This guy goes on: "The Intel Management Engine is a subsystem composed of a special 32-bit ARC microprocessor that's physically located inside the chipset. It's an extra general purpose computer running a firmware blob that is sold as a management system for big enterprise deployments. When you purchase your system with a mainboard and Intel x86 CPU" - that is to say with an Intel chipset - "you are also buying this hardware add-on: an extra computer that controls the main CPU. This extra computer runs completely out of band with the main x86 CPU, meaning that it can function totally independently, even when your main CPU is in a low power state like S3, suspend.

"On some chipsets, the firmware running on the ME implements a system called Intel's Active Management Technology. This is entirely transparent to the operating system, which means this extra computer can do its job regardless of which operating system is installed." So it doesn't mean Windows. It could be Linux. It could be, you know, or like no OS, if it's just sitting there waiting to be deployed. But it gets worse.

"The purpose of AMT is to provide a way to manage computers remotely. This is similar to an older system called Intelligent Platform Management Interface (IPMI), but this is more powerful than that. To achieve this task, the ME is capable of accessing any memory region without the main x86 CPU knowing about the existence of these accesses." I mean, it is a classic hardware backdoor. "It also runs a TCP/IP server on your network interface, and packets entering and leaving your machine on certain ports bypass any firewall running on your system." This cannot be blocked by anything that you do running on top of it.

"While AMT can be great value-add, it has several troubling disadvantages. ME is classified by security researchers" - and this is what we talked about at the time - "as

Ring -3." You know, normal apps run at +3. Ring 0 is the OS. Well, this is way beneath the OS.

Leo: I didn't even know there was a ring that's a negative ring.

Steve: Ring -3. "Rings of security," he writes, "can be defined as layers of security that affect particular parts of a system, with a smaller ring number corresponding to an area closer to the hardware. For example, Ring 3 threats are defined as security threats that manifest in user-space mode. Ring 0 threats occur in kernel level. Ring -1 threats occur in a hypervisor level, one level lower than the kernel. Ring -2 threats occur in a special CPU mode called SMM" - that's System Management Mode - "a special mode that Intel CPUs can be put into that runs a separately defined chunk of code. And if attackers can modify the SMM code and trigger the mode, they can get arbitrary execution of code on the CPU." But that's the main CPU. And this is -3, even below that.

Okay. So Intel rates this, their own problem, as "critical remotely exploitable." I would love to know what external vulnerability this represents, but this is part of the problem. The information is so blacked out that I could find nothing about how to scan for it, how to detect it, I mean, like anything. But here's the concern is that systems will not update themselves. Unlike browsers and OSes and many apps, the BIOS doesn't. You normally need to go get it.

Now, people like Lenovo, who have tried to integrate system management, controversial as it is, I would imagine that, if you are using the Lenovo "keep your system up to date," I know that it does BIOS updates because I've been a longtime ThinkPad and then a Lenovo user. I imagine that they could use that mechanism to push that out. But if you've just got, in the last nearly a decade, any Intel system that has this IME, the Intel Management Engine technology in it, then we don't really have a way of gauging, you know, I don't want to run around, hair on fire, screaming that the sky is falling.

Leo: Well, I mean, there's mitigation. If you don't use the built-in network, but use your own network card, you're safe; right?

Steve: Correct, correct. Yes.

Leo: You know, it has to be these managed computers. It's in every Intel chip, but I don't - it's not - it's my sense it wasn't, the management engine wasn't enabled unless you have a managed system.

Steve: No, that's not correct.

Leo: That's not correct, okay.

Steve: It's absolutely, it is absolutely enabled. Now, one thing you can do on Windows machines, if you browse through your list of services, and I did this on my Win7 X1, and there it was, Intel Management Engine, a service running. And that's the other problem is that this is also vulnerable to local exploit, not just remote exploit.

Leo: So Intel says you have to have vPro technology for this, which is not all Intel chips.

Steve: Correct. Good. I'm glad you mentioned that. Yes, that is right. Wikipedia's page has already been updated. It was updated immediately to be current about this. They write: "Currently, AMT is available in desktops, servers, Ultrabooks, tablets, and laptops with Intel Core vPro processor family, including Intel Core i3, i5, i7," and the Xeons.

Leo: But that vPro was sold as an enterprise system. So, I mean, I'm sure it's on your ThinkPad X1 Carbon because that's an enterprise computer. But I bet you a lot of, I mean, I wouldn't assume - for instance, I have a Mac with a Nehalem processor. I would assume - or actually with a Xeon, as well. I would assume it's not enabled there because they're not vPro systems.

Steve: Well, okay. So we probably need to not use the word "enable." It's present.

Leo: Available, right.

Steve: Right, right, right. If it's present, it's on because we can't turn it off, unfortunately. But right. So maybe it's just not there. I've got a bunch of links at the bottom of this about...

Leo: What a mess.

Steve: There is something, you can google "Intel management engine verification utility." I found that.

Leo: Ah, there you go.

Steve: It's dated 2010, and Intel's page says it supports XP through Win7. I'm sure it runs on Win10 because Win10 will run things that Win7 does. So google "Intel management engine verification utility." It's a small little - it's about half a meg, 500 and some K, a zip file containing six files. You can run that. And so that will check your machine locally to see if you have the Intel Management Engine present. And it confirmed that my X1 Carbon did. How to Geek has an article, "How to Remotely Control Your PC Even When It Crashes," where they go...

Leo: But that's the point of the management engine.

Steve: Correct.

Leo: It's an enterprise feature designed for the IT department to manage your system remotely.

Steve: Right.

Leo: So it's not nefarious that it exists. It's just...

Steve: Oh, no, no. And I never meant to explain it. The problem is Intel was absolutely secretive.

Leo: Right.

Steve: And it was always a concern.

Leo: Always, especially in the open source community. They hated that this thing existed.

Steve: Right.

Leo: It's just that you don't, you know, you buy an open source system, you build open source, and you still have some proprietary blob that you can't examine.

Steve: Exactly.

Leo: And now all of their fears are proven true.

Steve: Correct.

Leo: Terrible.

Steve: And then the last link is, it's an Intel community site titled "How to Find Intel vPro Technology-Based PCs." And unfortunately it's not a simple way. But that link, also in the show notes, it takes you, I think there are like four different ways you can check your system to see whether it has vPro technology and thus has this ME component. And bottom line is it would be good to look for any firmware updates. It's not something we all do all the time. I mean, and even the firmware documentation generally says, unless you are actually having a problem that you know this firmware will fix, better just to leave well enough alone.

In this case, if you've got the problem - again, because it's so underdocumented, I can't gauge how exploitable this is. I don't know, for example, what traffic this is sniffing on the primary NIC on a motherboard to know whether, like, how a remote exploit would be affected. Because, for example, if it were some obscure port, and it had to come in on an obscure port, or if it was an obscure protocol, I mean, it could be anything that works in an Intranet. I don't even know if it's for sure that it's routable. That's the problem is it's just - it's a big black box. But now we know Intel is calling it a critical, remotely exploitable vulnerability.

Leo: Are there exploits that we know of?

Steve: No. As far as we know, it's been done. Now, I did run across anecdotal supposition. But again, that's just all it is. We call that now "fake news," I guess, in this era. But there are people who are claiming that this has been exploited, but without evidence. So I ignore that because Intel really did try to tighten this down. I believe you have to have a certificate that the Intel Management Engine recognizes. So traffic needs to be signed. It is encrypted and secured with TLS. So, I mean, it looks like the bar is likely very high.

On the other hand, Intel has called this critically remotely exploitable. So hopefully, I mean, it's good news that they're finally responding. I wish I could gauge its true exploitability for our listeners, but there's just no information. So I will certainly be looking for more. This just happened yesterday. So again, as you said, Leo, even though a number of researchers have been saying to Intel, you know, tell us about this, and several people have been saying - in fact, at the very end of this report I have a site, SemiAccurate.com, which is not the most encouraging...

Leo: Only half accurate, yeah.

Steve: Not the most encouraging domain name. And the page is remote-security-exploit-2008-intel-platforms is in the URL at SemiAccurate.com. And this guy just rakes them over the coals, saying that he's been pounding on them forever to, like, fix this, telling them, and they've just been ignoring him.

So anyway, I have some new 1U Intel servers that I will be deploying. The one that I've already got in place has got its own external physical hardware firewall because it's where I will be bringing up the public GRC forums to support SQRL. They're actually - they've been online for almost a year now, but I haven't taken them public yet. And so it's already protected.

But, I mean, it is an Intel motherboard, and it's a state-of-the-art chipset, so I'm sure it's got this. And I will be updating its BIOS before it sees the light of day. And it's just - it's just frustrating for a researcher to be kept in the dark by something that looks like it's really important because how do you mitigate this if you know nothing about it? But all we can do is respond by updating our BIOSes. And, wow, hopefully - I don't know. This needs to be a lesson somewhere about the danger of absolutely black, black boxes.

Leo: Well, it's an opportunity at this point for AMD, and I hope AMD takes advantage of this. There's been some encouragement for AMD to make sure that their new Ryzen processors are coreboot, libreboot compatible. They'll have - they don't have this management engine, don't have any unknown blobs. And I think a lot of people, certainly in the free software space, would be jumping onboard. Could be very good for AMD. I would - that's disappointing. Disappointing, yeah.

Steve: Yeah. So, okay. Speaking of disappointing, five researchers at the University of Michigan have published their research, which went public last week during the IEEE European Symposium on Security and Privacy. The research paper, and I've got a link in the show notes to the - yeah, it's like a 17-pager. It goes on and on - but titled "Open Doors for Bob and Mallory."

I should mention we talked about Alice and Bob as being the standard characters that are used to, sort of in schematic form, to talk about two parties communicating securely. Well, Mallory, by convention, is the man in the middle, thus man and Mallory, so "M" for Mallory. And so this is "Open Doors for Bob and Mallory: Open Port Usage in Android Apps and Security Implications." And I'll just share their abstract where they pull all of this 17 pages down into the kernel, but summarizes beautifully what they found.

They write: "Open ports are typically used by server software to serve remote clients" - of course we know that - "and the usage historically leads to remote exploitation due to insufficient protection." You know, can anyone say "Mirai botnet"? Of course that's open ports, or Windows printer-and-filesharing.

"Smartphone operating systems inherit the open port support, but since they are significantly different from traditional server machines in performance and availability guarantees, little is known about how smartphone applications use open ports and what the security implications are. In this paper, we perform the first systematic study of open port usage on mobile platforms and their security implications. To achieve this goal, we design and implement OPAnalyzer, a static analysis tool which can effectively identify and characterize vulnerable open port usage in Android applications.

"Using OPAnalyzer, we perform extensive usage and vulnerability analysis on a dataset with over 100,000 Android applications. OPAnalyzer successfully classifies 99% of the mobile usage of open ports into five distinct families, and from the output we're able to identify several mobile-specific usage scenarios such as data sharing in physical proximity. In our subsequent vulnerability analysis, we find that nearly half of the usage is unprotected and can be directly exploited remotely. From the identified vulnerable usage, we discover 410 vulnerable applications with 956 potential exploits in total." So just shy of a thousand exploits.

"We manually confirmed the vulnerabilities for 57 applications, including popular ones with between 10 and 50 million downloads on the official market, and also an app that is preinstalled on some device models. These vulnerabilities can be exploited to cause highly severe damage such as remotely stealing contacts, photos, and even security credentials, and also performing sensitive actions such as malware installation and malicious code execution. We have reported these vulnerabilities and already got acknowledged by the application developers for some of them. We also propose countermeasures and improved practices for each usage scenario.

"To get an initial estimate on the impact of these vulnerabilities in the wild, we performed a port scan in our campus network and immediately found a number of mobile devices in two minutes which were potentially using these vulnerable apps. We've reported these vulnerabilities to the relevant parties through the vulnerability tracking systems" - and they've now got registered CVE and CERT registrations. "Some of them have been acknowledged. We encourage readers to view several short attack video demos." And there's a site that I have the links for here in the show notes.

So finally, under their "Threat Model," there are three ways these can be attacked. They said: "The threat to an app with open ports comes from the attackers with the ability to reach these ports. In the design of popular smartphone operating systems such as Android, ports are reachable from both the same device, for example, another app or a script on the web page, and another host in the same network with the victim device. Thus, compared to the majority of previously reported smartphone app vulnerabilities that only consider the threat from on-device malware, open port apps additionally face threats from network attackers," in other words, the local network attacks and web attackers, meaning malicious scripts, which is much more diverse and also of wider range.

"More specifically, in this paper we consider the following three adversaries: Malware on the same device, a local network attacker, and malicious scripts on the web." And I'll just note that they write: "When a victim user visits an attacker-controlled website using their mobile device, malicious scripts running in the handset's browser" - or delivered through an ad - "can exploit the vulnerable open ports on the device by sending network requests, which doesn't require permission. For each of these three threat models, we have prepared short attack video demos on our website to help readers more concretely understand their practical exploitation."

And so if we harken back to Firesheep, remember that the scenario there was in an unencrypted WiFi environment such as, for example, the often-used Starbucks example. What Firesheep allowed was it was doing promiscuous sniffing of all the network traffic and parsing the nonencrypted, that is, the HTTP transactions. Any query that, for example, a browser that was at the time, back then, logged into Facebook using HTTP, in order to maintain the persistent login, the session cookie was sent with every browser query back to Facebook. So a passive sniffer of network traffic could grab that cookie and log in, essentially clone the logged-in session by itself sending that cookie, and they would be logged in as someone.

So that's what Firesheep did. It was called Firesheep because it was a plugin that ran on Firefox that just - it was freaky. If you ran Firesheep at Starbucks, down the left-hand column would come up the identities of people surrounding you in the coffee shop, and you could click on one of them and be logged in as them. So those days changed because pretty much now all of those major services are HTTPS exclusively, and so all of those, all of that traffic is encrypted.

What this means - so relative to that, now we have a situation where Android apps running on Android devices are in many cases opening up listening ports for whatever reason. And they're often open and left open and are vulnerable. So a port scan within a nonencrypted WiFi environment like Starbucks will find those open ports and can often identify the apps and then exploit them. In other words, we're always talking about how our contemporary desktop OSes now always have a firewall between the OS and the external Internet. And so we have that line of defense.

Then we also have a NAT router that we're behind. So that allows machines in the private network to communicate with each other. But both the local firewall and the NAT router protect us from the public Internet. The concern here is that it turns out, and what these researchers found, is a substantial number of Android apps are opening ports which are then vulnerable to local scan. And in order to provide their functionality, they are not firewalled. Now, you'll still be protected within your local network by the NAT router that bridges you to the public network, but this is still a large attack surface.

So anyway, I'm glad these guys did the research and have brought it to everyone's attention. I once - I'm sure I talked to you, Leo, about how annoyed I was that iOS didn't make it easy for me to move things back and forth between my Windows-based desktop and an iOS device. I found an app which is exactly like this, which brings up a web server or an FTP server - actually in this case I use it as an FTP server. And I've got it now on several of my iOS devices.

I, of course, because I am very security conscious, I always make sure to turn it off when I'm not using it so that it closes that port. But it's very handy. I fire it up. Like if I want to just grab a big photo, for example, without having to email it to myself because I don't have iMessage under Windows so I can't message it to myself. So I'll turn this little server on, put the photo in it, and then I have a shortcut that allows my browser to immediately bring up the FTP server that the iPad is now hosting, and I can just click on it and suck the file right over.

And so this is one of those apps. There is an Android app that is a WiFi filesharing utility. And the problem is that port is open and makes this app vulnerable. And as we know, on the first level is access to the server. But then what we find is that the servers are not themselves secure. So even if the server was password protected and trying to only offer limited things, unless, I mean, I would argue, unless it's been pounded on, it's almost sure to have some mistake made that could convert this into either denial of service, crashing the app, maybe the phone, depending upon where the server is running, how deep in the kernel the service is, or maybe give someone access to even more data. And these researchers apparently were able to do that.

So I guess the takeaway for our listeners is, if you're aware that you are running any device like this, that is, any app in an Android-based device, be sure to turn it off. Disable it when you're not actively using it, very much the same way we've always been saying turn off Bluetooth unless you need it on. It's been a constant annoyance that every time Apple updates iOS they turn it back on again. And so it's like turning it off if you just don't need it saves power and also reduces your attack surface.

And for people who are a little more technically savvy and curious, there are, and we've talked about them before, local port scanners, which - so you could, if you looked at - you could turn your Android device on, determine what its local LAN IP is. That would be a 192.168 dot something dot something, typically. And then, from a different machine, you could run a local port scan, scan all 64K, you know, 65535 ports of that IP from your machine and see if you find anything open. The scan shouldn't take long, but it would give you an idea of what ports that Android device has open.

And then what's important to remember is that those ports are open wherever you go. So as you roam around, and the device connects to various networks, all of those ports are exposed. So doing a local port scan of an Android smartphone probably, this research would suggest, is very worthwhile. If nothing else, if it shows nothing open, then you have the peace of mind of knowing that. But if it surprises you, you'll want to find out what apps are opening what and for why because, if you can reach them from a machine through WiFi, anybody else can, anytime you're connected to a nonencrypted network. And probably even a network you have to log into, although it wouldn't be as easy as doing a passive scan. You have to work, bad guys have to work a little harder. But this looks like potentially big nugget.

Leo: All right, Steverino. I have your audio lined up for whenever you want that, by the way.

Steve: Okay. We'll get to it in a few minutes.

Leo: Yes, yes.

Steve: Okay. So we've had some fun talking several times about how IoT devices are in some cases, I guess "ratting out" would be the expression, their owners or their users. We had that case where there was some strange death in a hot tub, and the owner of the house claimed to have no knowledge of what was going on, yet his IoT-enabled water meter showed some huge amount of water consumed between 2:00 and 3:00 a.m., presumably to wash the blood away. And so as a consequence of that - and of course we've also talked in terms of law enforcement wanting to get subpoenas for anything that the Amazon Echo device may have overheard and so forth. Well, we have another one.

Leo: This one's wild. This one is wild.

Steve: It is so bizarre. And this is why I was saying at the top of the show that law enforcement officials, presumably the police who showed up to respond to a 911 call, had to have just been eyeing each other, like how dumb does this guy think we are?

So as I understand it - and I'm not going to go through the whole story. I've got the link here. Sophos covered this in their Naked Security blog. And the punchline is that a man's wife was murdered, shot by his .357 Magnum, which he had purchased some time before. Their marriage had apparently been under stress for some time. They had two sons. They weren't getting along. Apparently he was taking money from her accounts. He had a girlfriend on the side, and he said to her that he would be leaving his wife. When the police arrived, responding to the 911 call which he placed, one arm and one leg were zip-tied to the chair, or a chair, and his other arm, as I pictured it, was somehow zip-tied like up to his neck. And it's like, okay, is it not obvious that he did this to himself? But I guess, you know...

Leo: I can't move. I can't move. Well, except for this arm. And this leg.

Steve: And he tells a story about a large, 6'2 hooded camo-wearing intruder who was the perp behind all of this skullduggery. Anyway, the point of all this is that - so he outlines all of this. And then the police tap into social media and look for any other evidence. And it turns out that his wife was wearing her Fitbit because she was planning on going out, I think it was yoga or exercise of some sort that morning. And so it, of course, recorded, as the Fitbit does, all of her movements. And the timestamped record that they were able to recover contradicted the husband's version of events by more than an hour. Like there was a complete disparity in the timeline.

So anyway, yes, I guess people attempting to perpetrate crimes are going to have to be very careful about what IoT devices, whether they're baby monitors or Fitbits or anything which is monitoring the environment. Not as easy as it once way to get away with stuff.

Leo: No, and there's cameras everywhere; and, golly, it's just a lot harder than it used to be.

Steve: So one of our constant themes on the show is the problems with interpreters, how surprisingly, but almost understandably, difficult it is for interpreters to be secure. The media interpreters, image interpreters, all of these fancy formats we have are interpreted, meaning that even the old TIFF, the tagged image file format, well, it's composed of modules with tags which label what the module contains. And so an interpreter displays a TIFF image, or a PNG, or a GIF, or a JPEG, or an MPEG, or an MP3. The MP3 is a compressed - it's compressed by having a representation of the audio, which is then an MP3 player reads that representation and reconstructs an audio approximation of the original sound.

So a classic 28-year-old interpreter, I mean, not quite as old as SpinRite, but it's been around 28 years, since 1988, is Ghostscript. And it turns out that there are some serious problems in Ghostscript. The security advisory that I could find - this has just happened, so it hasn't percolated out through all of the various places where Ghostscript is in use. But, I mean, it is the go-to standard Postscript and PDF interpreter, which reads those high-level page descriptions and converts them into a raster image for display or printing. And so the specific vulnerabilities that I saw affected essentially all recent versions of Ubuntu from 12.04 LTS all the way up through 17.04. So if you're an Ubuntu user, the danger is, as we've often talked about with Adobe PDFs - there's another classic. Adobe Reader, how many years of material has this podcast had thanks to Adobe's PDF problems?

Well, turns out Ghostscript may be entering the same sort of zone of having lots of problems. A researcher, Kamil Frankowicz, took a close look at the latest release and discovered multiple significant vulnerabilities: Ghostscript improperly handles parameters to the rsdparams and eqproc commands, which allow an attacker, exactly as was the case with Adobe's PDF Reader, to deliberately craft malicious documents that could disable OS protections and thereby allow and enable execution of arbitrary code; or, if they're not quite slick enough, cause a denial of service, that is, cause the application to crash.

He found use-after-free vulnerabilities in the color management module of Ghostscript, which could also at least cause a denial-of-service application crash. He found a divide-by-zero error in the scanned conversion code in Ghostscript, which an attacker could again leverage, and multiple null pointer dereferencing errors which, again, could be leveraged for attack.

So this is the advantage of open source. As we know, open source doesn't automatically magically make something more secure. But it at least enables someone to examine the code and find problems. This, of course, is the problem with IME, Intel's Management Engine, is they've chosen to keep it closed and protected to an insane level so that you can't even reverse engineer it. It's all encrypted, and it decrypts on the fly into RAM in order to run in this hidden 32-bit ARC processor. So there's, like, here's two different examples.

So neither open nor closed specifically says whether or not it's secure. But with something like Ghostscript, which is open source, if somebody takes the time to look at it, to carefully read the code - and the people who wrote it can't do it. It's just it's a fundamental law of the universe. You just cannot see errors in your own code. You have to have somebody else look at the code, wanting to find problems, and they just reveal themselves to somebody who's sufficiently skilled in finding these kinds of security vulnerabilities, you know, a Tavis Ormandy sort of guy.

So anyway, I would say Ubuntu users be on the lookout for updates to Ghostscript, and anyone else, probably any variant of Linux. Debian had a problem with a licensing change because Ghostscript has been taken under the wing of someone who has moved the Ghostscript code from pure GPL to a variant of that, that I know that Debian had a problem with. But I would imagine it's still prevalent. And it's not clear whether these were newly introduced, or whether they've been longstanding. But, I mean, it is the case that the oldest Ubuntu, 12.02 LTS, was subject to this. So I would just say to our listeners, if you know that you've got Ghostscript around, if you're a Linux user and you're able to display PDFs, certainly Ghostscript, which is the display interpreter, unless you're running a version of Adobe's PDF for Linux, it's worth checking out.

Following up on our podcast from a few weeks ago on all the various ways that information leaked, we talked about what a VPN could do, what cookies were doing, first-party versus third-party and so forth. Somebody a few weeks ago - and I put this in my notes, and unfortunately I can't give credit to him because it got away from me. But thank you for reminding me that Server Name Indication (SNI), which is an extension that was added to SSL and TLS way back in 2003, is yet another way that where we are going on the Internet can escape. We've talked about SNI in the context, not of a privacy concern, but as a feature enhancement in the past.

The way servers traditionally operated, they wanted to bring up HTTPS connections - or generically, more broadly, SSL or now TLS connections - is at the server side you would bind the server's security certificate to an IP so that any connection on that IP would be secured under the certificate bound to that interface, to that IP. So essentially the IP represented that domain that the certificate covered. The problem, of course, arose where you wanted multiple hosting. You wanted multiple domains, all served from a single IP.

So in order to accommodate that, an extension needed to be added to the client hello packet, that is, the first packet going to the server after the TCP, the underlying TCP connection is brought up. Then the client, like typically the user's web browser, it sends the client hello packet, which among other things lists all of the SSL or TLS protocols that it supports. And remember, so that allows the server to look at the ones it knows and hopefully choose the best one from among those that they have in common. And so then the server hello goes back to the client, saying this is the one I've chosen. And then the client hello also has a nonce, a big random number, and the server also chooses one. And so they exchange their nonces, and that allows them then to negotiate a key and so forth.

Well, part of the, in the updated spec, part of the addition in the client hello, the first packet the client sends, is the SNI, the Server Name Indication, which now all browsers, even the popular WGET, WGE Tool, command line file retriever tool, it knows about it. Everything knows about it for years now. It's been in all browsers for up to 11 years. That packet has to be in plaintext. It is not encrypted. It can't be encrypted because it's before the encrypted tunnel is brought up. In fact, that packet, and by reading the Server Name Indication out of the client hello, a multiple domain, a multiply hosted server can then choose which certificate to respond with based on the domain name that is in that packet. So unfortunately it's unencrypted, can't be encrypted. It's in plaintext. And anybody sniffing traffic, even when everything else is encrypted, an ISP - we did talk about how an ISP could even, if you were encrypted, could see what your IP was.

Well, it turns out that, yes, not only that, but in the client hello packets which are labeled brightly so that servers are able to understand them, an ISP could capture those and look at the SNI, the Server Name Indication, in every outgoing client hello and see which domain you're going to. So they didn't even have to do reverse DNS, and they don't have a problem with multiply hosted sites, not knowing which site you're going to at an IP, because the client hello tells them.

So I just wanted to add that, too, and thanks to our listener for reminding me that Server Name Indication is yet another way that our privacy leaks, despite everything we want to do. And there's no solution for that. All you could do would be to VPN yourself past your ISP's view, and then let your traffic out onto the Internet in some public location where you're not worried about it being captured and looked at.

Okay. Now, this is the week of bad problems for some reason. Mayday. Yeah, this is really bad.

Leo: It's so bad he can't even say it.

Steve: Microsoft Edge browser has a vulnerability, believe it or not, that is not patched, that was not disclosed responsibly, that allows arbitrary sites that you visit to steal your cookies and passwords for other sites.

Leo: Well, that's kind of a flaw.

Steve: I know. That's why I'm breathless. A serious same-origin-policy bypass. We've talked many times about how crucial it is that browsers honor the same-origin policy. That's the idea that code running, like that came from a certain domain, cannot just arbitrarily go look at some other domain. It's able to go make other requests of its own domain, that is, of its own origin, the same origin, but not others. So the sandboxing, within-origin sandboxing, is crucial.

So this security researcher, he, like, focuses on browser security. And in fact his domain is called BrokenBrowser.com. He's found in the past some more than 500 vulnerabilities in browsers and gleefully reports them. His name is Manuel Caballero, based in Buenos Aires. And apparently he has no interest in responsible disclosure. Doesn't seem to be any - no concern given to it. He has posted proof-of-concept exploits. He's got videos demonstrating it.

This affects Microsoft's premier Edge browser, for which there is no current patch. And again, I'm sort of speechless. This vulnerability can be exploited to allow an attacker to obtain a user's password and cookie files for their other online accounts. It leverages some mistakes Edge's developers made in the handling of so-called "domainless pages" such as about:blank. About:blank is a domainless page. And it turns out that there have been problems in the past that Edge has fixed, and this guy found another one, a way around the previous fixes. So it feels like there's a fundamental architectural mistake that was made in the Edge's design which they have now been patching because this just shouldn't be a problem. And he keeps finding other ways around it.

So versions of proof-of-concept demos are hosted online at his site at BrokenBrowser.com. And since you may not - and, I mean, they actually work. So if you go there with a Microsoft Edge browser, you can have your other sites, your Facebook and Google and Amazon and so forth, password and session cookies shown to you. But since you probably don't want to do that, he has also posted video demos which are available. And all of this is now in the public domain. He notes that the vulnerability can be customized to dump the passwords or cookies of any other online service, including Facebook, Amazon, and others. The flaw affects only Edge because universal cross-site scripting and same-origin bypasses, as he writes, tend to be specific to individual browsers. Well, thank goodness for that. But this is still bad.

Because, as we know, modern ads deliver JavaScript code to browsers, attackers can leverage malvertising campaigns to automate the delivery of this exploit to thousands of victims or more. Manuel explained that attackers are able to use malvertising to push their malicious code into cheap banners shown on popular sites. If an attacker is hosted inside a Yahoo banner, and the user is logged into their Twitter account, that user can be owned with no interactions at all. And this is true, and he demonstrates it.

So this is bad. This is like the flipside of responsible disclosure. It's unfortunate that this is now out in public, and apparently there's no fix. The only thing I could suggest people do, I mean, if this is a concern, because there doesn't seem to be a workaround, I mean, look for workarounds. I will certainly mention it next week.

Leo: Or don't use Edge, would be the workaround.

Steve: I would say yes, not use Edge. I would switch to Chrome and stay away from Edge until this gets fixed.

Leo: Although, since that's the primary rendering engine on Windows 10, I wonder...

Steve: I know.

Leo: ...how often it gets used without your intention. Probably in email; right?

Steve: Yes, yes, good point. Very good point. So where are we? This is Tuesday the 2nd. So our May Patch...

Leo: [Crosstalk].

Steve: Yes, our May Patch Tuesday will be next Tuesday. It would be wonderful if Microsoft had time to fix this by then. This has to have lit a fire under them because this is really bad. And it sounds like it should be an easy fix. Whatever it is this guy found, I would imagine they can patch their way around it and get it fixed. And let's hope they do that.

So Chrome has made a change to the way they handle certificates that I thought was interesting. They are deprecating in Chrome 58 their use of the subject name field, which sort of - or it's called the subject common name, which is the way the certificate identifies itself. The problem is that the way the common name was originally defined was never very rigorous. For example, I think I've got www.grc.com as my main certificate's common name. But I noticed that, ever since I started using DigiCert, they also placed it, that is, the www.grc.com, in the SAN field, the subject alternative name.

And traditionally the subject alternative name, as its description sounds, are alternative names.

And this is, for example, how you can get one certificate from a certificate authority which can be used on, like, three different domains, where for example I might have GRC.com, www.grc.com, and media.grc.com. You would have the common name would be one of them, and then the alternative names would be the other two. But I noted that DigiCert, again, they're on top of their game, they were always putting all three in the subject alternative name.

Well, it turns out that the industry has been souring over time over the use of, like the actual value in the common name as standing for something. So for a while Chrome's behavior was to prefer the subject alternative name field; but, if it was missing, then to fall back to using the common name, assuming that the common name would be the domain name that was bound in the certificate. Although, again, it was never rigorously defined. The format wasn't defined. It's considered an untyped field, which makes everybody nervous. Whereas the subject alternative name, being more recent, was rigorously defined from the start.

Anyway, so the change is significant. That is, with 58 the fallback path is removed. And from now on, so this represents really a change for the industry, the common name will still be in certificates. But browsers moving forward, and Firefox is on the same trajectory, will no longer be using the common name for any use other than display purposes in the certificate. The browser will no longer rely on that. So the reason this could affect individuals is that a popular thing to do is to generate self-signed certificates. And many of the original self-signing tools do not support subject alternate name fields. They only self-sign. And the domain that you're signing is in the common name. Well, users of those certificates will suddenly discover that Chrome no longer honors those self-signed certificates.

The good news is there are newer tools for generating fully compliant certificates, but that may mean that people who have long-expiration self-signed certificates are going to need to create updated certificates, which Chrome from 58 on, and soon Firefox, will continue to honor. So just sort of an interesting, again, something that Google, I think, is doing the right thing in doing in continuing to move forward and clean up a little bit of what's been done before.

We have a mystery botnet that's got a lot of people wondering what's going on. It's a new IoT botnet that's being called the "vigilante botnet" which has been growing rapidly. It was first spotted in October of last year, of 2016. It's also known as Hajime, H-A-J-I-M-E.

Leo: Hajime [crosstalk] botnet.

Steve: Hajime botnet. What's puzzling people is that it is extremely well designed and sophisticated, with a robustness and feature set that surpasses its overtly malicious rivals like the Mirai botnet, for example. It is expending a huge amount of effort to infect other IoT devices. But unlike Mirai, once Hajime affects an IoT device, it closes the backdoors behind itself, securing those devices it has infected against further attacks. It blocks access to ports 23, 7547, 5555, and 5358, which are known to be the most widely used vectors for infecting IoT devices. And thus, at least temporarily, it sanitizes that device in its wake.

But rather than using the more common fixed command-and-control server architectures that we have been talking about, Hajime employs a decentralized peer-to-peer network to issue updates to infected devices, making it far more difficult for the botnet to be taken down - I would argue impossible - by anyone. And what's also strange is that, when infected devices are also equipped with display terminals, every 10 minutes or so it displays a signed message describing its creators as, quote, "Just a whitehat securing some systems."

Leo: Uh, yeah.

Steve: And then it says, "Important messages will be signed like this. Hajime Author. Contact closed. Stay sharp."

Leo: Uh-huh. Uh-huh.

Steve: It's like, okay. So unlike Mirai and other IoT botnets, Hajime lacks DDoS capability.

Leo: I like that you've adopted my pronunciation, Hey Jimmy. Hey Jimmy.

Steve: Hey, Jimmy.

Leo: Hey, come on over, Jimmy.

Steve: ...lacks DDoS capabilities and other hacking skills or capabilities, except for the propagation code that lets one infected IoT device search for other vulnerable devices and then infect them. Kaspersky's security researchers noted that, quote: "The most intriguing thing about Hajime is its purpose. While the botnet" - it never gets old. "While the botnet is getting bigger and bigger" - now, we're talking 300,000 infected devices at this point, by the way.

Leo: Wow. Wow. That system has a big network.

Steve: A third of a million, "partly due to new exploitation modules [so it's evolving also] its purpose remains unknown." Kaspersky says: "We haven't seen it being used in any type of attack or malicious activity. Its real purpose remains unknown."

And Radware's write-up provided some additional interesting technical details. They said: "The distributed bot network used for command and control and updating is overlaid as a traceless torrent on top of the well-known public BitTorrent peer-to-peer network, using dynamic info hashes that change on a daily basis. All communications through BitTorrent are signed and encrypted using RC4 with public and private keys.

"The current extension module provides scan and loader services to discover and infect new victims. The efficient SYN scanner implementation scans for open ports on TCP port

23 and TCP 5358. Upon discovering open Telnet ports, the extension module tries to exploit the victim using brute-force shell login, much the same way Mirai did. For this purpose, Hajime uses a list consisting of the 61 factory default passwords from Mirai and adds two new entries, 'root/5up' [root is the username, 5up is the password] and 'Admin/5up' [admin is the username, 5up is the password] which are factory defaults for Atheros wireless routers and access points. In addition, Hajime is capable of exploiting ARRIS modems using the password-of-the-day 'backdoor' with the default seed as outlined here.

"Hajime does not rashly follow a fixed sequence of credentials. From Radware's honeypot logs they were able to conclude that the credentials used during an exploit change depending on the login banner of the victim." I mean, this thing is top-drawer. "In doing so, Hajime increases its chances of successfully exploiting the device within a limited set of attempts to avoid the system account being locked out, or its IP being blacklisted for a set amount of time. Radware also suggested that the flexible and extensible nature of the Hajime botnet would allow it to be used for malicious purposes, including conducting real-time mass surveillance from Internet-connected webcams. However, since Hajime has no persistence mechanism, as soon as the infected device is rebooted, it goes back to its previously unsecured state, with default passwords and the Telnet port open to the world."

Now, there's no evidence of this, but I wouldn't be at all surprised if this is, I mean, I would be happy, in a way, if this was the NSA. That is, here we have a situation where Mirai brought down DynDNS last year because so many of these IoT devices were infected that a hugely powerful, what was it, it was 600GB or something, a ridiculous amount of traffic was able to be generated. So now along comes a botnet which you cannot take down, that uses high-end, torrent-encrypted, BitTorrent system intercommunication with rotating password-of-the-day seed-based passwords that protects the devices it infects from subsequent infection, stays in RAM, doesn't hurt them, doesn't destroy them, but takes them essentially out of service and is, due to its architecture, incredibly hard to kill.

This feels like a well - I mean, and at the same time is a massive surveillance network, should the owner of this network choose to exploit these IoT devices. We've never seen any evidence of this. And reverse engineering of the implant demonstrates all it does is rapidly find other vulnerable devices, presumably those that have been recently rebooted, and reinfects them with it before anybody else can find them, in order to essentially cleanse this otherwise very worrisome IoT install base of this latent problem. This, to me, this feels like a state actor who's solving this IoT problem for us.

Leo: So does it give you any clues that Hajime is Japanese. It's the first name of a boxing manga series, comic book series. Hajime no Ippo.

Steve: Well, and I don't know who named it.

Leo: Ah. Maybe they saw that text string in there or something like that.

Steve: Could be. I did not find any reference to where the name came from.

Leo: Anybody in the chatroom speak Japanese? Does Hajime mean anything in Japanese? Hmm.

Steve: Oh, I do know what it means because it looked it up this morning when I got the pronunciation. But now I don't remember.

Leo: I like Hey Jimmy. Hey Jimmy. Ha, hey Jimmy, how are you? Good to see you.

Steve: Yeah. Ugh. So we're all familiar...

Leo: It means "beginning" in Japanese.

Steve: Ah, interesting.

Leo: It's a Japanese martial arts term. That's scary.

Steve: Yeah. It is, a little. So we're all familiar with the concept of a Breathalyzer, where if law enforcement pulls you over when you're driving, and believes that maybe you're intoxicated, traditionally you could blow into this device, and it would register your current blood alcohol level as indicated by the alcohol content in your expiration. Well, NPR reports that legislation has passed in New York which may pave the way for a "textalyzer."

They write: "If you're one of the many who text, read email, or view Facebook on your phone while driving, be warned: Police in your community may soon have a tool for catching you red-handed. The new 'textalyzer' technology is modeled after the Breathalyzer" - except it's not - "and would determine if you had been using your phone illegally on the road. Lawmakers in New York and a handful of other cities and states are considering allowing police to use the device to crack into phones because, they say, too many people get away with texting and driving and causing crashes."

I'm going to skip a bunch of this reporting, which is about a personal story of somebody who was involved in a distracted driver crossing the center line and causing an accident that resulted in a death and how difficult it was to generate probable cause to get a subpoena to pull the records. But skipping down, it says: "Even though New York and most other states ban texting and other kinds of cell phone use while driving, [this individual] Lieberman says those lawsuits are difficult to enforce. The takeaway is our current law is a joke. Lieberman, along with the advocacy group he cofounded, has been working with a company" - and we know them, Cellebrite, which are of course the people who...

Leo: Oh.

Steve: Uh-huh.

Leo: They make the cell phone suckers.

Steve: Exactly. "Cellebrite has been developing the textalyzer. It would be able to determine whether a driver illegally was using a phone in the moments before a crash. Cellebrite engineer Lee" - and here's a name, Papathanasiou, P-A-P-A-T-H-A-N-A-S-I-O-U.

Leo: Yeah, that's Greek, Papathanasiou.

Steve: Papathanasiou.

Leo: Opa!

Steve: Papathanasiou.

Leo: Papathanasiou.

Steve: "...demonstrated the device for lawmakers" - who could finally pronounce his name - "and reporters at the New York State Capital in Albany earlier last week. He says a police officer just goes to the driver and attaches a cord to connect the device to the phone. The driver doesn't even have to let go of the device. Papathanasiou said: 'They simply tap one button. It will process for about 90 seconds or so, then display what the last activities were - again, that could be a text message and so on [but also web activity and touchscreen use] - 'with a timestamp.' The device would display a summary of what apps on the phone were open and in use, he says, as well as screen taps and swipes. 'For example, if it was a WhatsApp message or a call, it will indicate what the source was, the timestamp, and then what the direction of the communication was, so if it was an outgoing call versus an incoming call.'

"Papathanasiou says the technology still is not yet fully developed, but would be tailored to what's legal in each jurisdiction that approves its use. And he insists that the textalyzer would only capture taps and swipes to determine if a driver was using the phone, that it would not download content, and that it would be able to tell if the driver was using a phone legally, hands-free. In New York, the bill authorizing police to use the textalyzer has passed out of one committee and is pending in the next. Lawmakers are interested in the device in New Jersey and Tennessee, and in Chicago as well as other cities, as they consider ways to get drivers to focus on the road instead of their phones."

So I'm a little suspicious of whether they can pull this off because they've demonstrated success in getting into phones, but at least at the moment that's not something that the phones support, although this is one of those things where it might not be surprising to see that kind of technology officially enabled in phones for exactly this kind of purpose. I don't know. But we do know that this distracted driving is a real issue. I look at cars weaving on the road, and we've talked here many times about how lights will turn green, and no one's car moves.

Leo: Right.

Steve: They're all taking a timeout, and they go, oh, oh, oh, and then off they go. Yeah.

Leo: Wow.

Steve: We know we have SaaS, S-A-A-S, software as a service. Now there is RaaS, R-A-A-S, ransomware as a service. Yes, for just $175, "a new ransomware service on offer from a Russian-speaking user is reputed to be a boon to less technically capable cybercriminals. Going by the name Karmen, K-A-R-M-E-N, anyone can deploy this easy-to-use drop-in ransomware kit, without any need to understand how it works. The security firm Recorded Future posted last week that a Russian-speaking user called DevBitox [D-E-V-B-I-T-O-X] has been advertising the ransomware in underground forums."

Karmen, the name of this ransomware, "is part of a worrisome new trend known as Ransomware as a Service. It allows less technically skilled amateur hackers with little technical knowhow to inexpensively purchase access, in return for which they receive a complete suite of web-based tools to develop their own ransomware attacks. In Karmen's case it offers an easy-to-use dashboard interface. Buyers can modify the ransomware, view what machines they've infected, and see how much they've earned." Yeah, I guess it was inevitable. But ransomware is going to be with us for the foreseeable future.

And finally, the FCC has announced its plan to reverse Title II's enforced Net Neutrality. The Verge reported, and this has got heavy coverage: "The Federal Communications Commission is cracking open the Net Neutrality debate yet again with a proposal to undo the 2015 rules that implemented Net Neutrality with Title II classification. FCC chairman Ajit Pai called the rules" - that is, the rules enacted in 2015, so we have a new FCC chairman in the Trump administration - "'heavy-handed' and said their implementation was 'all about politics.'" I guess that may be true. "He argued that they hurt investment and said that small Internet providers don't have 'the means or the margins' to withstand the regulatory onslaught." Okay.

"Ajit Pai said last Wednesday: 'Earlier today I shared with my fellow commissioners a proposal to reverse the [what he called the] mistake of Title II [classification] and return to the light touch framework that served us so well [as he put it] during the Clinton administration, the Bush administration, and first six years of the Obama administration.'"

The Verge writes: "His proposal will do three things: First, it'll reclassify Internet providers as Title I information services," which I happen to think is a mistake. "Second, it'll prevent the FCC from adapting any Net Neutrality rules to practices that Internet providers haven't thought up yet. And, third, it'll open questions about what to do with several key Net Neutrality rules, like no blocking or throttling of apps and websites, that were implemented and put in place in 2015.

"Pai said the full text of his Net Neutrality proposal would be published" - and presumably it was last Thursday afternoon because this statement came out last Wednesday. "It'll be voted on by the FCC at a meeting on May 18th." So a little more than two weeks from today. "From there, months of debate will follow as the item is opened up for public comment." And I think all of us listening need to take advantage of this and make sure that our representatives in Washington know what we feel about this. "The commission will then revise its rules based on the feedback it receives" - let's hope that's true - "before taking a final vote to enact them."

And then The Verge continues: "Strong Net Neutrality rules were passed in 2015 and have been in place for about two years. Those rules reclassified Internet providers as 'common carriers' under Title II of the Telecommunications Act, which subject them to tough, utility-style regulations." Amen. "The FCC has previously mandated under Title II that Internet providers follow a few key rules: no blocking of sites and apps, no throttling the speed of sites and apps, and no paid fast lanes. The rules applied to both wired and wireless Internet providers and also gave the commission oversight of 'interconnect' agreements between Internet providers and big content companies like Netflix. Internet providers have, of course, been unhappy about this, as they'd rather not have the FCC looking over their shoulder and limiting what they're able to do with their network. They sued to overturn the rules, but so far the rules have held up in court. But that may not last."

So once again, we have all of us little folks who are trying to keep the Internet open, yet we have major, huge, well-financed, deep-pocket ISPs that say they need the flexibility to do with their traffic what they want. They're wanting to be considered information providers rather than common carriers. And to my thinking, maybe a solution would be for them to bifurcate. If they want to do some things, then split themselves into half so that there's a common carrier portion that carries the traffic, and then an adjunct that can offer content. But mixing these things together is just a recipe for trouble. And I just hope there is, yet again, another very loud outcry for keeping things as they are and keeping our ISPs regulated under Title II.

Leo: Yeah, we fixed this last time. Tom Wheeler wasn't going to do this, the chairman of the FCC at the time. And he opened it for comment, and literally more than a million comments.

Steve: Yup.

Leo: And it convinced him of the merit of choosing Title II. That and the President's urging that he do so. I don't - I think we're in a different climate, so we'll see what happens. But we can make those millions of comments again. I think we need to.

Steve: Yeah, it's absolutely worth doing, to re-voice our position.

Leo: Yup.

Steve: Cloudflare, who as our listeners know has become a sponsor of the TWiT Network netcasts, has launched a new service. And I don't completely understand it, but I'm sure we will with time. It's called Orbit. You can find out - there's a short write-up at www.cloudflare.com/orbit, O-R-B-I-T. And it is a new service to protect IoT devices. In their write-up they said: "Technology is changing, shifting towards a world where low-cost connected chips power products used by billions of people around the world. Everything from jet turbines and oil rigs to cars, cameras, and clothing are coming online. And while these tiny chips unlock incredible potential, they are a liability if not secure." To which I add, "Amen."

"When PC vulnerabilities are discovered," they write, "software vendors issue a patch, which end-users are required to download and install. These patches keep PC software up-to-date and secure. IoT devices also require patches, but the PC security model cannot scale to 22 billion devices. IoT manufacturers often haven't built over-the-air update mechanisms and are terrified that updates will brick a user's device. In the meantime, consumers never think about having to upgrade their Internet-connected 'toaster,'" Cloudflare has in quotes.

"Cloudflare Orbit solves this problem at the network level by creating a secure and authenticated connection between an IoT device and its origin server. Orbit takes the Internet out of IoT. Behind Orbit, devices are" - and then they have "I*oT."

"Orbit allows device manufacturers to instantly deploy 'virtual patches' and block vulnerabilities across all devices on the network simultaneously." In other words, Cloudflare is imposing - and this is me speaking - Cloudflare is imposing itself as a proxy server network in between all of these IoT devices and the public Internet, or the server that is intended to serve the IoT devices that exists on the public Internet.

And then going back to what they say: "This keeps malicious requests from reaching devices, buys time for IoT manufacturers to carefully QA their updates, and keeps devices from leaking data or launching DDoS attacks." And I should mention also that that also allows them - that allows Cloudflare to put up firewall rules instantly to close ports or filter traffic, which are discovered to be vulnerable for IoT devices. And it uses mutual authentication with client-side TLS certificates. We of course are always talking about server authentication, where the server contains a signed certificate that asserts its identity. They're suggesting the use of client-side certificates.

Now, this of course is questionable because you want to keep your certificates private. And it's not clear how an IoT device can inherently keep its certificate private. For example, GRC and any public server goes to great lengths to keep its private key, which is in its certificate, private. So I don't know how you enforce that. It's an increase in security.

A spokesman for Cloudflare said: "Orbit sits one layer before the device and provides a shield of security, so even if the device is running past its operating system's expiration date, Cloudflare protects it from exploits. And while devices may be seldom patched, the Cloudflare security team is shipping code every day, adding new firewall rules to Cloudflare's edge. Orbit" - and this is something I didn't know, but it's already in place. "Orbit has been built in collaboration with a number of IoT vendors and already protects over 120 million IoT devices. It allows IoT companies to write logic on Cloudflare's edge and create firewall rules that are immediately updated to the Cloudflare Orbit layer for all devices, without having to write and ship a patch." And I noted that - is it Eeros? Eero, E-E-R-O?

Leo: Eero. Eero.

Steve: Those devices are being protected by Cloudflare Orbit.

Leo: Oh, nice. Another sponsor of ours.

Steve: Yeah.

Leo: All the sponsors are working together. I like it.

Steve: Yeah. So this is not something that can be added afterwards, that is, this needs to be - an IoT vendor would need to decide they want to take advantage of this service. And so they would work with Cloudflare to - so essentially Cloudflare sort of becomes a CDN for IoT firmware and also an Internet proxy for the IoT device to hide behind so that it creates - essentially, Cloudflare is giving IoT vendors their scalability in order to provide useful services and features to Internet-connected devices. Which I think sounds like a really cool idea. So it's not something that an end-user needs to worry about. But it's something that IoT device manufacturers - it's a service they could use with Cloudflare to enhance the security of their IoT devices.

Leo: Fantastic.

Steve: So cheesy things aren't going to bother. But devices that care about security and are wanting to use this as a value-added benefit could avail themselves of this.

Leo: Nice.

Steve: So bravo to Cloudflare and those device vendors who choose to use it.

And I just did want to mention that Mozilla and Chrome are continuing their back-and-forth with Symantec. I read through a mind-numbing dialogue of we said this, and they said this, and then they proposed this, and we read that, and then we proposed that. Symantec is, of course, pushing back as fiercely as possible and wanting to do as little as possible. And this of course is, as we've discussed in previous podcasts in response to rather gross violations of the responsibility of a CA, which Chrome and Mozilla have both decided they're going to take action, that they're not going to let this stand. Things like removing EA certificate issuance completely, enforcing short certificates moving forward until Symantec proves and takes clear measures to demonstrate that they will be responsible moving forward and so forth.

So browsers, the browser vendors, are attempting to both appear and be understanding and reasonable, while also feeling that their true responsibility lies with their users, who are inherently trusting their browsers to keep them safe. So, I mean, this is one of those situations where there's going to be pain. And now there is an ongoing struggle to decide where the line gets drawn. So I just did want to mention this is ongoing. I'm kind of keeping an eye on it, but we don't have any conclusions yet.

Some bits of errata: Irwin Wessels shot me a tweet saying "There are not individual emojis for each skin tone variety. They're modifiers/ligatures." And so that prompted me - so thank you, first of all, Irwin. That prompted me to dig in a little bit. And believe it or not, we now have emoji racial diversity. It is in the unicode spec under "diversity," if you can believe that. Of course we had the original sort of Smurf yellow emojis. Those were the ones I always used. And I misstated last week that now we had, like, an explosion of emojis. But apparently the unicode space was big enough to accommodate them. And we also talked last week about, yes, there are, what, 17 million, more than 17 million - wait, no, it was 17 planes of 65K. I don't remember now how many million. But there was, like, plenty of unicode space.

Well, it turns out that what was actually done was that a skin tone modifier was added to the unicode spec quite some time ago, in Unicode v8, back in mid-2015. And so we still have the original Smurf yellow, but then we have five human skin tones ranging from white through dark, kind of light white to dark brown. And that's an escape character, essentially, which can be appended to any of the existing Smurf yellow emojis to turn them into one of five skin tones. So thank you for the correction, and I find that kind of interesting.

Oh, also I misspoke when I referred to the EternalBlue and Eternal, you know, the various Eternal* things, and DoublePulsar, as being Vault 7, thus CIA leaks. I meant to say, and they are from, the NSA's Equation Group, which was released by Shadow Brokers. And I think you actually corrected me, Leo. But somebody else noted that, as well. Actually, several listeners, so thank you for that.

And then this is not quite errata, but this is under the category of "There's got to be a simpler way to say that." Joel Dittmer quoted me regarding punycode. He said: "Classic @SGgrc." Apparently I said: "I don't disagree that this was never not a bad idea." What? I don't disagree...

Leo: Double negative.

Steve: ...that this was never not a bad idea. Is that a double or a triple? Anyway...

Leo: Never not a bad idea is a good idea.

Steve: I don't disagree that this is never not a bad idea.

Leo: It was always a good idea is never not a bad idea.

Steve: So, so there. I'm not sure what that means. Anyway, under Miscellany, I did want to point our listeners to BadSSL.com, a cool and quick little website that checks your browser client for its feature set, shows whether it supports, continues to support old things that it should not, whether it supports new good things. You need to - it's not quite obvious when you bring up the page. You need to click on something in order to make it run the tests. Then a bunch of things spin around, and then it shows you what your browser is doing. So BadSSL.com. And at the bottom of that page they remind us about that our friend Ivan Ristic, also over at SSLLabs.com has a very nice client-side SSL test with a number of different features, as well.

I talked about the idea of bringing up a VPN on Amazon's EC2. And I just saw, as we were starting the podcast, that there's a formal project for that. But there is that, but there's also you're able to set up a proxy server so that you can run all of your stuff through an Amazon EC2 VPC instance. So I've got a - I just wanted to make a note of that. And the link for the steps to do that, just 10 steps you run through, is on GitHub.

Several people said, after my mentioning ChromaZone, the very first thing I wrote for

Windows, the tool that I used to teach me how to program Windows, Jame, I guess it's Jame Bong, said: "I desperately need to play around with ChromaZone. How do I get it?" Many other people said the same thing. I'll publish it all publicly. I'll put that page back online and put links to the code so people who can set a machine to 256 color mode can play with it. I'd be happy to have people do so. I'm really proud of it. It's sort of my masterpiece of Windows programming. And now, Leo...

Leo: I'm ready.

Steve: Two minutes of technically completely accurate - this is not gibberish as the Turbo Encabulator was. And I know our listeners will love this because it is technically accurate description of missile guidance. But it is wonderful. So everybody listen up.

[Audio Clip]

MALE VOICE: The missile knows where it is at all times. It knows this because it knows where it isn't. By subtracting where it is from where it isn't, or where it isn't from where it is, whichever is greater, it obtains a difference, or deviation. The guidance subsystem uses deviations to generate corrective commands to drive the missile from a position where it is to a position where it isn't; and arriving at a position where it wasn't, it now is. Consequently, the position where it is is now the position that it wasn't, and it follows that the position that it was is now the position that it isn't.

In the event that the position that it is in is not the position that it wasn't, the system has acquired a variation, the variation being the difference between where the missile is and where it wasn't. If variation is considered to be a significant factor, it, too, may be corrected by the GEA. However, the missile must also know where it was.

The missile guidance computer scenario works as follows: Because a variation has modified some of the information the missile has obtained, it is not sure just where it is. However, it is sure where it isn't, within reason, and it knows where it was. It now subtracts where it should be from where it wasn't, or vice versa. And by differentiating this from the algebraic sum of where it shouldn't be and where it was, it is able to obtain the deviation and its variation, which is called "error."

Leo: It's just algebra. Very straightforward.

Steve: Yes. So anyway, I loved how pedantic that is. And it's technically correct, so...

Leo: It's accurate. It's just 1 minus x2, yeah.

Steve: In the show notes, and I tweeted this, is something that does not work over the podcast, over an audio podcast. But believe me, it is so good. Four minutes and 10 seconds, the Cookie Monster, the Muppets' Cookie Monster consuming a machine as it describes itself.

Leo: Somebody said this is actually Ed Sullivan. It's that old.

Steve: Oh. And I had no idea that the Cookie Monster could be so descriptive.

Leo: Yeah, yeah. Yeah, this is "The Ed Sullivan Show," I gather.

[Video clip]

Leo: Yeah, you really should watch the video. Jim Henson. That's pre- "Muppets Show," I think. He's eating it.

[Video clip]

Leo: He's eating it. All right. You get the idea. I encourage our viewers to [crosstalk].

Steve: It's so good. It's just fabulous.

Leo: Yeah.

Steve: And it's got a punchline. It keeps going, and it gets better. So really it's worth it. I tweeted it. And again, google "Muppets analytical computer." If you google "Muppets analytical computer," you'll find it.

Leo: Wow.

Steve: Another listener tweeted: "Got the first Frontier Saga book. Let me tell you, wow, amazing, and I haven't been able to put it down." And that echoes the sentiments I'm getting from many of our listeners, so I'm glad to have another recommendation that people are enjoying. And I heard you refer to it also over the weekend.

Leo: Yeah, John's been reading it. I bought the cheap three volumes on Kindle, although I notice it is available on Audible, so if I like it I might go to the audio version.

Steve: I think you played it, and it was a little nasally for you.

Leo: Yeah, yeah.

Steve: You didn't like the...

Leo: That's right, yeah.

Steve: You didn't like the guy who was reading it.

Leo: I'm enjoying reading it. Occasionally I need to read letters on paper, or something like paper.

Steve: Yeah.

Leo: Yeah.

Steve: Brent Longborough is a long-term Twitter finder and sender. I recognize his name. Anyway, he said: "Hi, Steve. After years of shameless freeloading off a friend's SpinRite, today I purchased my own copy. Thanks and apologies." Brent, no apology needed. You're legit, and I'm glad to have you. And you'll be able to play with 6.1 during its early pre-release stages, as will everyone who has 6 at the time. And, gee, I don't know how to pronounce this, Engrpiman said: "@SGgrc SpinRite saved my database server. RAID 1 disk failed. Used SR to bring drive back to life. Made backup, got new drives, restored from backup."

Leo: Wow. That's like a whole novel in 140 characters. A beginning, a middle, and an end. Conflict, resolution, the whole thing.

Steve: Got the whole thing.

Leo: The whole thing. I'm thinking Engineer Pi Man.

Steve: Oh, nice.

Leo: But I don't know. It's like reading license plates; right?

Steve: Yeah, it is, yeah.

Leo: E-N-G-R-P-I-M-A-N.

Steve: And then four little bits of feedback from our listeners closing the loop. John, @Mr._John_Morris, said: "Listening to SN-607 and thought I'd share the reality of Chrome cookie settings, not respecting settings." And then he sent me a photo where actually it was two side-by-side photos where he showed he had disabled cookies in Chrome, or third-party cookies, or first, you know, some cookies, and then went over and looked at them and found them.

And I was familiar with this. I wrote back to him, and I wanted to share with our listeners, some browsers will continue to store, but not to send cookies. Some will even continue to receive an update, but not store or not - I'm sorry, but not send cookies. This is exactly why I created that Cookie Forensics page because, if you look there, you'll notice it shows like how stale the cookies are. We were talking about this a couple weeks ago when we were talking about stale cookies. Because back then when I created it, sometimes you could turn off cookies, and some browsers would send the cookies they had, but wouldn't update them newly. So then they were stale. Some browsers continued to update them, but didn't send them.

So the fact that the browser still has the cookie and didn't delete it doesn't mean that it's still sending it, but it may just have it. And then, if you were then to reenable cookies, it would immediately start sending the cookies that it had. The point is this is very complex, potentially very complex and unintuitive behavior. But the Cookie Forensics page, because it actually runs through multiple cycles of exchanging cookies back and forth and analyzes the whole set of transactions, it's able to weed everything out and then summarize everything for exactly the way the browser is working. So you can't - it's not something you can statically inspect. You need to see how it actually acts in vitro, or vivo, wherever that would be.

Richard Hardy tweeted: "Watching the Security Now! DoublePulsar episode, I had a thought about the port 445 being open. What about PCs in a DMZ?" And so that's a very - so the point was I was wondering, how could there be so many systems with 445 open? And he notes, well, if you set up a DMZ on your router, by definition unsolicited incoming traffic is not dropped, it's sent to that IP. And it's true. If you then had a machine that either didn't have its own local firewall running, and probably 445 would have to be a Windows machine or a Linux machine that had Samba running on it, if it didn't have a firewall or did have 445 open, then, yes, it would be vulnerable to the SMB.

The good news is all Windows systems since Windows XP SP2 have had a software firewall in and enabled by default. So even that would protect you unless something on the machine had poked a hole through it on purpose. So again, the DMZ is worth noting because that does certainly create a vulnerability to the machine that is the recipient of unsolicited traffic, very much like having it on the public Internet. But at least Windows machines where the 445 port problem is the most acute probably are protected themselves.

Joan tweeted: "There are ads related to my location popping up on my Facebook feed. How do I stop this if setting up a VPN isn't enough?" And Joan, I guess I would refer you back to our podcast a couple weeks ago about the real privacy protection, the second one, where we actually finally made time to get to that, because this is the problem is that even if you VPN, then where you go still knows who you are because that's conveyed through your cookies. So what you would need to do would be to turn on incognito browsing and VPN so that your IP would change, and your browser would then respect your privacy in whatever your browser's version of incognito mode is. Then you should not be known.

So those two things, turning on incognito mode and use a VPN. Suddenly your IP will change and your browser should then, I mean, maybe fingerprinting would still be a problem with the browser. I'm not sure, browser by browser, how good a job they do. But they'll at least block cookies so that you are no longer logged in, and the places you go don't obviously know who you still are, even though you're coming from a different location. Which is the only thing a VPN really does for you is just shift your location to where the VPN's public presence is.

Leo: You can also, in many cases, a browser has a setting for location sharing. You should do that and everything you just mentioned.

Steve: Correct.

Leo: Because if the browser's giving up your information, it kind of doesn't matter what you do.

Steve: Right. That's a very good point. I've got mine set up, I don't know if it's the default, but mine sometimes [crosstalk].

Leo: It usually asks; right? Yeah.

Steve: Yes, and they prompt, you know, do you want this site to know where you are?

Leo: You can in the settings then go look and see who's got permission and who doesn't. And you can change the settings, say never do it, yeah.

Steve: Nice. And lastly, finally, Nate G. says, actually in a pair of tweets: "The biggest roadblock with getting friends and family onboard with a password vault solution has been the master password. Any suggestions?" And he continued: "My wife, for example, sees the benefit of them, but has had problems remembering a high-entropy password in the past. Unwilling to try again now." And so, you know, empathizing with him, I wrote back and I said: "Nate. Best advice would be to use maybe, for example, five memorable real words with one deliberate misspelling. Not the best solution possible, but not a bad compromise."

Leo: I'll tell you what I do.

Steve: Okay.

Leo: That I find very easy, and I tell people to do this. If you have a poem memorized or a song lyric memorized, let's say "The Jabberwocky." "Twas brillig, and the slithy toves did gyre and gimble in the wabe." You don't want to use the words, but you could use the first initials of each word.

Steve: Right.

Leo: Perhaps adding punctuation and uppercase and lowercase letters, depending on some arbitrary rule that you conceive of.

Steve: Or maybe even a comma where there would be a pause.

Leo: That's what I would recommend. I don't want to say what I do, but that's what I would recommend. And then, to make it long and strong I then add numbers, either say a child - some number that you remember. You know, the one I always tell people is the childhood phone number because that's something that probably is not on record anywhere, but it's something you also memorized as a child; right? That was like the most important - or your childhood zip code or address. That gives you an additional padding. And the two combined I think would be fairly strong.

Steve: I think that's good. And I'll bet that, after a while, you begin to memorize it. I've got some gibberish that I've been using for, you know, in safe places for quite a while. And I just type it without even thinking now.

Leo: It was the best of times, it was the worst of times. You know, people memorize stuff. And there's usually a phrase or two. And what's the length? I would say 15 to 20; right? Just keep going until you have 20 words and then add a phone number.

Steve: I would say shorter than the Canadian national anthem.

Leo: You heard me trying to remember that. Well, but there's a good example. If you remember the national anthem, maybe not yours, some other country's national anthem, that'd be a perfect example because in your head you could actually sing it and type your password. Not the words, just the initials. I think the words might be a little bit brute-forceable. I don't know. Do you think bad guys have initial brute-forcers?

Steve: I think that in general the password vaults are, as we know, are generally safer against brute force.

Leo: Yeah.

Steve: For example, we know that LastPass does a strong, what is it, 500 or more now...

Leo: A thousand, I think.

Steve: ...iterations of PBKDF.

Leo: PBK - yeah.

Steve: And so it makes it difficult to brute-force it.

Leo: To brute-force it even if it were a bad password; right?

Steve: So it's certainly, yes, so it's certainly better to use a bad password with a password vault than, like, I mean, this makes me wonder how good the passwords Nate's wife is using in general are.

Leo: Yeah.

Steve: She's probably using, like, not, I mean, the vault allows you to just reduce it to a single one gnarly password. But frankly, the attack profile is such that it doesn't have to be, you know...

Leo: That gnarly, yeah.

Steve: That gnarly, yes.

Leo: Okay. That's good to know, yeah. And then turn on two-factor, and then it's really gnarly; right? Because...

Steve: Yup.

Leo: Yeah.

Steve: Yup, that would be perfect.

Leo: LastPass supports that, so that's - I always do that, too. Steve, we're done.

Steve: Oh, my goodness.

Leo: That was a quinti venti latte day if I ever...

Steve: Big bad news week. Wow.

Leo: Lots of it. But we'll do it again. How about this? Do you think there'll be enough to do a show next week?

Steve: Oh, I'm afraid there will be.

Leo: Patch Tuesday?

Steve: If nothing, following up on some of these nightmares.

Leo: We do the show every Tuesday, 1:30 Pacific, 4:30 Eastern, 20:30 UTC. Stop by. Say hi. Watch live. It's everywhere. On our website, TWiT.tv/live. YouTube has it, YouTube.com/twit. Ustream, Twitch, we all have TWiT channels. And then join the chatroom, too, irc.twit.tv because that's like the cool kids in the back. And there's a lot of, you know, you don't watch the chatroom. I know you've been in the chatroom many times, but you don't watch it during the show. But there's always lots of stuff, you know, comments. It's great. Research.

If you can't watch live, on-demand audio and video is always available at our site. And Steve's got audio and, uniquely, he's got human-transcribed transcriptions of the show. Elaine Farris does a great job. So you can go there and read along with Steve. And you know, while you're there, pick up a copy of SpinRite. It wouldn't hurt you to give Steve a little money, support him: GRC.com. He's got a lot of free stuff there, too. And then you wouldn't feel guilty using that and listening to the show; right? And we'll be back next week.

Steve: And you'll be able to get the pre-release of 6.1 as it's coming along, before everybody else.

Leo: And is it coming along?

Steve: As soon as SQRL's behind me. And we're making great progress on SQRL.

Leo: Good, good. He's going to get that SQRL back. If you're just tuning in, don't worry, just hang out with us a while, it'll all make sense.

Steve: SQRL. SQRL.

Leo: SQRL. SQRL. Thanks, Steve. We'll see you next time on Security Now!.

Steve: Thanks, Leo.