



FREAK and RowHammer

Description: Leo and I catch up with several VERY interesting security events and stories of the week. Then we take a deep dive into two of the week's big security stories: FREAK and RowHammer.

High quality (64 kbps) mp3 audio file URL: <http://media.GRC.com/sn/SN-498.mp3>

Quarter size (16 kbps) mp3 audio file URL: <http://media.GRC.com/sn/sn-498-lq.mp3>

SHOW TEASE: It's a little early in the morning, but Steve Gibson is here. He's back. He's got a little bit of a cold. But we're going to talk about two major exploits, you've probably read about them in the headlines, FREAK and RowHammer. We'll also look at Microsoft's Patch Tuesday update and a lot more. Security Now! is next.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 498, recorded March 12th, 2015: FREAK and RowHammer.

It's time for Security Now!, the show that protects you and your loved ones online, in this case a very early edition. Hey, Steve Gibson.

Steve Gibson: Hey, Leo. Great to be with you, a little bit later than usual in the week, but here nonetheless.

Leo: You almost made it 500 shows without getting sick.

Steve: Oh. Oh.

Leo: Steve never gets sick. And I guess you got a bad - sounds like a cold.

Steve: It's funny, I realized I've sort of forgotten how.

Leo: How to be sick?

Steve: Yeah.

Leo: You probably don't have any cold medicines in the house or anything.

Steve: No. There's sort of a protocol, like okay, now I do this and this. It's like, I was, like, rediscovering. It's like, okay, well, that's interesting.

Leo: So we missed our Wednesday. And by the way, even though we made great pains, put it on the front page of the website, I tweeted it, you tweeted it, still people said, "Where's my Security Now!?" People love the show. And so I'm glad to say, even though it's late, you will get a Security Now! this week.

Steve: Absolutely. So, yeah. Some tweets I saw said, well, there goes my Wednesday commute.

Leo: I know. People are hooked.

Steve: It's like, okay, well, you know, how about Friday? You can wrap up the week with a Security Now!. And as it turns out, the fact that you were able to run MacBreak Weekly long on this particular Tuesday was fabulous because you had a really great panel, and it was a marathon, a three-hour marathon.

Leo: It worked out well. It worked out well. We did a three-hour, which we couldn't have done. So, yeah, no, it was actually serendipitous. However, here we are.

Steve: So this was set to be a Q&A, but two really important and interesting things happened that just can't be covered briefly, or shouldn't be. And that is something called FREAK, which is an acronym for something, doesn't really matter what, and RowHammer, which is a surprising, recently surfaced weakness in DRAM which - I guess I would characterize RowHammer as not really a side-channel attack; but, if you really, really, really, really, really, really want to exploit a system and have no other means, this might work. So...

Leo: Sounds the kind of thing the NSA really likes.

Steve: Yeah, I was going to say exactly that because the details are specific to the processors and chips. And so it's the kind of thing where the NSA might say, "Oh, look, he's got one of those laptops, so let's hammer him." So anyway, so that's a - so today's podcast will cover those two things in detail. And then we'll do the news of the week first. So I think everyone's going to be satisfied with our slightly delayed podcast.

Leo: Sounds excellent to me.

Steve: We'll talk about Patch Tuesday, which happened. The news that the CIA has been dogging Apple's encryption since 2007, since before the iPhone, and has been putting a lot of pressure against, like, into cracking the iPhone.

Leo: Yeah, we didn't talk about it on MacBreak Weekly because I thought, oh, we'll talk about this on Security Now!.

Steve: Yes.

Leo: It's really better for Security Now! since this is, you know, a security issue.

Steve: Yeah. And then some odds and ends about routers and redirection. I did want to check in with you about the Apple Watch, which I have a few comments about. And then we're going to plow into these really two big security stories, deep in technology.

Leo: This is great. We're loaded for bear, baby. Even though it's 8:00 in the morning where we are, I hope you have - back East they're thrilled. Hey, there's a show at my time, when I'm awake, yeah.

Steve: So we had a Patch Tuesday. Last Tuesday was Patch Tuesday. Not anything really significant, although Microsoft did fix, in this Patch Tuesday - one of the problems with this FREAK exploit got fixed on Tuesday, on March 9th. So that was good. So there were 14 bundles. Five were criticals, and all of those were remote code execution vulnerabilities. And the remaining nine they just flagged as important.

There was the standard cumulative security update for IE, which it's the standard monthly update. And with the boilerplate from Microsoft saying the most severe of the vulnerabilities could allow remote code execution if a user views a specially crafted web page using IE. So, you know, go to a site and get owned. So that's been fixed. The second one was a VB scripting problem. And I got kind of a kick out of this because they said - this is Microsoft saying "resolves a vulnerability in the VBScript scripting engine in Microsoft Windows. The vulnerability could allow remote code execution if a user visits a specially crafted website which uses VBScript." And I thought, okay, so nobody does that. What website has VBScript?

Leo: I don't know.

Steve: Maybe something inside Microsoft.

Leo: Or Intranets, I bet, yeah.

Steve: Yeah. Well, but, okay, yeah, right. In a Windows-based company. But why would you, in this day and age, not use JavaScript, which has become the industry standard for client-side scripting? So this is Microsoft saying, well, we produced something a long time ago, and we needed to keep it going, so we fixed a vulnerability that you really don't care

about because you're not using it.

Leo: Did you see that they made a "Time to break up with IE8" website now? I think it's...

Steve: Oh, my god.

Leo: I think it's Microsoft. It's...

Steve: So it's gone from 6 to 7 to 8.

Leo: Yeah. So it says, "Break up with Internet Explorer 8 on your six-year anniversary." So I guess that IE8 is six years old. And it's pretty funny because they're having people tweet their little breakup stories. "I'm leaving you because at least five of your action bars are not mine." I just love it. And then there's a little heart cursor that goes around. It might be their Valentine's Day thing. And then there's a button that says, "Oh, god, no, I'm not ready." And you click that, and it shows a video of "Internet Explorer encountered another problem and needs to close." It's pretty funny.

Steve: Now, this is by Microsoft?

Leo: Well, they've done these before, remember, because of the insecurity. Let me just see, if I press the "Dump" button, if it goes to Firefox or - "Text a breakup to IE8."

Steve: I just think this is very classy, and not the sort of thing you typically see from staid old Microsoft.

Leo: No, they did something cool for IE6, remember, because they were really concerned about...

Steve: Oh, my goodness, yeah.

Leo: Yeah. I don't - here's a WTF button. Let's see what - "Break up with IE8. For whatever reason, IE8 recently increased in browser share. Join the intervention and stop supporting IE8." You know, it might be the Firefox - it's really not apparent who this is from. "A public service, lovingly crafted by humans." It might not be anybody. It might not be Microsoft. It just is. But I love it. It's well done.

Steve: It's really well done.

Leo: Yeah. Wasn't Stuxnet fixed again?

Steve: Yeah. Yeah. Apparently there was like a multi-year-old patch that prevented the Stuxnet infection.

Leo: Oh, oh.

Steve: And then a workaround was found.

Leo: Oh, lord.

Steve: So you could still get your PC infected with the original Stuxnet nastyware. So, yeah, that's, again, we've got - the big revelation of the last couple years, and we'll talk about this in the next story about the CIA and Apple, is we've lost our virginity in this industry. I mean, it used to be just, you know, we gave them different colors of hats, white hat and black hat, and maybe gray hat sometimes, hackers. And we thought those were the people, the only adversary that we had. Come to find out that for, like, since the beginning, the U.S. three-letter-initial law enforcement and intelligence-gathering agencies have been absolutely frantic and furious and, unfortunately, well staffed and well funded to be every bit as aggressive, if not more so, than actual bad guys.

I mean, only recently we're seeing where this sort of use of malware is an actual profit center. For example, with the new file encryption technology that really does induce people to pay the ransom, previous to that viruses, as annoying as they were, they were annoying. You know, they got into your system and sort of mucked things up. But it was always sort of, oh, okay, why? They must just be doing it because they can. Well, what we know is that law enforcement and intelligence gathering are doing it because they feel really put upon, really endangered by the growing use of encryption. And it was foreseeable that encryption would grow.

And in fact we'll be talking about, in this FREAK attack that had people freaked out late last week and early this week and is still a problem, it is entirely because of the NSA's paranoia over cryptography that this FREAK attack is possible. And we'll talk about the export-grade ciphers which the U.S. government enforced on the industry in its early days because they wanted to weaken crypto only enough that it was strong enough to protect people kind of in general, but weak enough that they could crack it. And it was funny because, by calling strong crypto a "munition," which is like, what?

But now I understand that maybe better now than I did before. By calling it a "munition," by categorizing it that way, you could restrict its export. And what that meant was that within the U.S. we could use these so-called "munitions" peacefully with each other to get strong encryption. But that meant automatically that the encryption strength had to be downgraded for any connections leaving the U.S. And that meant that international communications could be cracked. So that was the rationale behind this notion of encryption as a munition is it's like, we don't allow their export, so we're going to call them a munition. You can't make a connection outside the country with strong crypto.

And so that also tells us that the NSA has been tapping the 'Net also for a long time because that's the other part of using crackable crypto is sucking in all the data so that

you can decide what you want to crack. So in that vein, more Snowden documents have come to light, published this time by, I should say "again," by The Intercept, because we covered the release of Snowden documents from The Intercept just recently.

Leo: The Intercept is that thing that Pierre Omidyar started. He hired Glenn Greenwald away and created this, I don't know what it is, for-profit journalistic enterprise called The Intercept.

Steve: Yeah, well, someone's being paid by the word.

Leo: Well, and also a lot of people left The Intercept, or at least some people left, unhappy with the way it was being run, I think.

Steve: Boy.

Leo: Is it really long?

Steve: It is redundant and long. You get teases up at the front. And then it's like, well, and then halfway down they kind of come back to that and add a little embellishment. I mean, it's like classic, how can we bloat this story to the largest size possible? I was, I mean, and I'm very tolerant of that. Normally I don't notice that. But this was just like, oh, come on. So when I was putting things together, I just, you know, it was like, okay, well, I already read that earlier. And I read that earlier. And so, you know, tell us something you haven't already said in the first graph.

Leo: Some of the best editors have left, is what the problem is.

Steve: Wow. So here's what we know. And there's a lesson in this for our listeners that you and I have spoken of often. And it's a little creepy. And that is, we have discussed this before in the context of HDMI and DVD, within the life of this podcast. And that is, when we were talking about DVD encryption in the early days, it was obvious really that, if your household, your - I guess we don't even have them anymore. But if your entertainment system's DVD player had the ability to decrypt an encrypted DVD that you had stuck in it, and by definition it has to have the ability to decrypt it, then right there sitting on the shelf is a box vulnerable to reverse-engineering. And if you reverse-engineer it, then you know how to decrypt DVDs, too. That is, the Achilles heel in this system was it absolutely depended upon keeping a secret.

And, boy, talk about a widespread secret that they were keeping. It's no surprise that they couldn't. But if something that you have access to must withstand divulging its own secrets, then it's inherently insecure. Which is not to say, I mean, in the absolute sense. Sitting there in front of you is something that knows how to decrypt what you want to know how to decrypt. There are ways to ask it what it knows.

So essentially the CIA, the Snowden documents show us through a series of slides that the CIA has for, what would it be, eight years been actively financing lots of Lockheed Martin employees, who's like a large - 80% of Lockheed Martin's business is government

contracts. And so they subcontract out a lot of this stuff. So they've been financing continuous cracking efforts against the iPhone. There's an annual meeting or meet-up called "The Jamboree," which they have in order to pull all the various components together. And then they give each other slideshow presentations of the capabilities that they have managed to develop since the last Jamboree a year ago.

So the researchers have claimed over the course of these years - and some of this, as is the case, we have to remember, with the Snowden documents, is old news. So things like going to the iPhone 6 and the newer iOSes, we know that Apple has been proactively aggressively working to thwart, if not the CIA, then the hackers, because it is also a danger to Apple's users if the iPhone gets rooted and hacked, jailbroken, and malicious [indiscernible] is in there. Apple wants to protect its users. So in the process it's protecting them also from the CIA eavesdropping, against which law enforcement wants to push back strongly.

So the researchers claim to have modified Apple's Xcode development platform so that any application built with it, built with Xcode, which is like that's how you build applications, will automatically include backdoors. And infected apps can somehow invite or entice - entice was the word in the document, it's like, what? - can entice other apps to join in. Okay? Also another target and, I mean, like a threat, it's a little unnerving to hear stuff that we revere for security being labeled a threat by law enforcement.

But the TPM, the Trusted Platform Module, has long been a target of this group. And I have long wondered why it never really took off. And I have to wonder, sort of in a Machiavellian way, if on some level all of these agencies aren't really working not to have a secure platform created because it would make their job so much harder. So, yeah, you know, we got TPM years ago. BitLocker can lock to it. But, you know, and there are some OSes that boot sort of through it. But, boy, it sure has never met its potential. So anyway, they don't like it. They don't like anything that is going to keep them from getting the information they want.

So quoting from a small piece of this Intercept article, they said: "The Snowden documents do not address how successful the targeting of Apple's encryption mechanisms have been, nor do they provide any detail about the specific use of such exploits by U.S. intelligence. But they do shed light on an ongoing campaign aimed at defeating Apple's efforts to secure its products and, in turn, its customers' private data.

"In the top-secret documents, ranging from 2010 through 2012" - so again, they are a bit dated - "the researchers appear particularly intent on extracting encryption keys that prevent unauthorized access to data stored and firmware run on Apple products. In an abstract of a 2011 presentation at The Jamboree, the researchers noted that the intelligence community is highly dependent on a very small number of security flaws, many of which are public." In other words they're using the same hacks that hackers are to jailbreak phones. They have no, I mean, that's been the only means that the CIA, NSA, FBI, and so forth had of cracking an iPhone was basically the public hackers are doing the work for the intelligence firms or agencies.

However, in this abstract, after explaining that the intelligence community is highly dependent upon a very small number of security flaws, many of which are public and which Apple eventually patches, then they promised that their presentation could provide the intelligence community with a, quote, "method to noninvasively extract" encryption keys used on Apple devices.

Another presentation focused on physically extracting the key from Apple hardware. In other words, the first is a side-channel attack of some sort, maybe using EMF emissions

or power consumption. And actually elsewhere in this document those were specifically mentioned. And this physical extraction with a key, back in the early days of chip fabrication, that was known as "popping the top." You pop the top on the chip and get out your high-resolution electron microscope or just a very good microscope, and there in front of you are all of the traces that a chip designer can reverse-engineer the chip's circuitry from.

I mean, and so this brings us full circle to the problem that Apple faces. And that is, just as with a DVD or an HDMI screen, which again HDMI of course is encrypted transit video, in those cases the consumer is seeing the unencrypted contents, the unencrypted data, because it's been decrypted for them by something in the living room. Similarly, anyone holding, like, booting an iPhone and installing software is holding the device that is able to decrypt that encrypted stuff.

And so there is, I would argue, a limit to what Apple is able to do. I continue to believe, all the evidence is that, even if Apple didn't know about this, and maybe they didn't, or maybe they did have some clue that this was going on, they've been taking all the measures you could ask to increase the security of their device in order to raise the bar as high as possible in order to thwart this. But we should remember that we're holding the decryption device in our hand.

Leo: Right. I wonder, you know, this is what, 18 months in, now. It was June of 2013 that the Snowden leaks began; right? I mean, we're well into the leaks. And maybe we're getting to the bottom of the barrel here. And the only reason I say that is the Gemalto story, where Snowden claimed, or the papers claimed, I think it was Snowden, wasn't it, that...

Steve: Yes. And that was our previous mention just a few weeks ago of The Intercept. It was another story in The Intercept.

Leo: So this story claims that the NSA had hacked the keys of this SIM card manufacturer. Gemalto makes billions of SIM cards a year. They're in most of the phones in the U.S. But Gemalto then did an audit. Now, whether this is to be trusted or not, I don't know. But they did an audit, and they said, yeah, you know, we saw some NSA-style attacks. They never got past our office systems. Our keys are intact, and the systems were not - the integrity of the systems that encrypt the SIM cards was never at risk. Now, maybe Gemalto, I mean, I don't know who to believe here. But I'm wondering if we're starting to get to the point in these papers where this is stuff, it's a little more speculative.

Steve: Right.

Leo: You know what I mean?

Steve: Right, right. And certainly, why would they hold the best stuff to, like, a year and a half later?

Leo: Yeah, this is good stuff. This is juicy. Well, I'll give you an example, the Xcode thing. Now, that makes sense. If you really want to compromise a system, compromise a development tool that's used by everybody. But you get Xcode from Apple. And you would be a strange developer that would get Xcode from some third-party source. So the NSA or the CIA in this case would have to - I think this is a barrier that's pretty high - get into Apple's servers, modify Xcode in such a way that nobody noticed. I just don't think this is credible.

Steve: Or plant an Xcode change in a specific developer's machine.

Leo: That you could do. You know, retroactively.

Steve: Yes.

Leo: But on the other hand, Xcode is patched all the time. And, I mean, this is why, when you get GCC or other open source compiling tools, you always get an MD5 hash, and you're supposed to compare the two.

Steve: Right.

Leo: There are automated tools that will do that and protect you from hacked code. But if you're getting it - okay. So it'd have to be post facto, after you downloaded it.

Steve: I think, yeah, because nowhere in this paper did they talk about how they infected developers with the infected Xcode. They said we've got an Xcode that is like an auto backdoor installer. And presumably they were then solving that problem by installing these in specific developers' environments.

Leo: Which makes it much less of an attack.

Steve: Right.

Leo: I mean, it's a very specific attack.

Steve: Right.

Leo: I'm not saying, I mean, certainly these need to be paid attention to.

Steve: Yeah, we just don't know.

Leo: And as I said last week, no one is anymore questioning the validity of these documents. There are just too many; there's too much detail. These aren't made up.

Steve: Right.

Leo: Snowden didn't write this in his cabin in the woods.

Steve: Yeah, I mean, yeah, the slides are always rather sobering because it's like, wow, there's a vocabulary for this. There's this huge machine that is rumbling forward, working on taking privacy away.

Leo: That's the biggest takeaway, is not whether they've had success in one area or the other, whether they're managing to crack Apple or not, but just that they're, at least until recently, continually attempting this.

Steve: Right.

Leo: But if you think about it, if you're law enforcement, TPM, any form of encryption, any form of security is, if you're trying to investigate a criminal, going to thwart you. So of course it's seen as an enemy in that regard.

Steve: A threat, right.

Leo: I think they also value, you know, and this is the weird kind of schizophrenic nature of this, they also value it. And they're always talking about CERT, and all these other agencies are always talking about how important it is to encrypt and protect your privacy. President Obama said, yeah, everybody should have encryption. So it's this weird dichotomy.

Steve: As long as we can see into it.

Leo: Well, I don't even know if they say that. As long as we can see into the criminals' encryption I think is the subtext.

Steve: Right, right.

Leo: As a good and honest citizen, you should have the right to privacy. But you cross that line, we want to be able to see what you're up to. You can't fault them for that.

Steve: No.

Leo: But you can't fault us for trying to thwart it, either. Not everybody who encrypts is a criminal.

Steve: Exactly. So I got a tweet from a James Bennett Saxon, who was following up on our conversation about router firmware and how, at this point, you really just can't trust the firmware on especially a low-end commodity router. And so he tweeted: "@SGgrc I have FIOS with Verizon-supplied router. Can I wipe their scary firmware?" Now, I liked this because it raised a really good point, which is there are situations you're going to be in where you can't replace the firmware, for whatever reason, on the router that you're stuck with. I'm sure that Verizon's random router firmware for their FIOS offering is not going to accept a Linux build downloaded into it.

So what I wanted to remind people is that routers are stackable. And if you, for whatever reason, have cause to mistrust your actual border router, like in this case the Verizon-supplied router that interfaces to his fiber optics externally, or FIOS system, there's nothing to prevent you from connecting that router only to another router that you do trust. And then that router, that second router, the internal router, becomes the one that supplies your internal network. They stack very nicely, no downside at all. Some additional features, actually, when you think about, like, having maybe a less trusted network and a more trusted network. So I wanted to remind people that, if they're frustrated with the inability to change their firmware, you can just install a router inside that one that you do trust with running whatever firmware...

Leo: Oh, that's a good idea, yeah.

Steve: ...that you choose.

Leo: That's where you put your Astaro gateway.

Steve: Exactly. So I got a tweet - I was talking about my blogs, and I don't remember the context of that, but on the podcast recently I mentioned that I have two blogs at WordPress. Steve.grc.com is my personal one, and blog.grc.com is the corporate Gibson Research blog. And I started getting tweets from people that said, "That doesn't work. I'm getting SSL errors." Well, I fixed it. And I had actually intended not - I had intended to leave it bad, just for that purpose, so we could show it.

Leo: I can show the tweet. I can show the tweet, I guess.

Steve: Yeah. But I loved what this turned out to be. I mean, when I saw these tweets begin to come in, it's like, huh? What? And so...

Leo: Are you hosting this, or is it a WordPress.com hosted blog?

Steve: Exactly.

Leo: Ah.

Steve: It is WordPress. And what I'm using, there are two ways that I could use my own domain name to allow people to get to the blog. And one is by doing a so-called "301 Redirect," where anyone who comes to blog.grc.com, that would go to my server. And my server would respond to the browser with a 301 permanent redirect, meaning that this website is saying that, to get the content for the domain you're asking about, blog.grc.com, go over here. And so then I would redirect the browser to WordPress.com/ - and I think, I don't remember now what the username is, I think it's AgileSynapse, which was very briefly my first Twitter handle before I decided that shorter is better.

Leo: What?

Steve: Yeah. The problem is...

Leo: You can use Steve Gibson, you know.

Steve: Oh, that was long gone.

Leo: Oh, really.

Steve: Oh, lord, and I feel sorry for Steve Gibson on Twitter. I'm constantly seeing him replying to people saying, no, I'm not that Steve Gibson. It's like, oh, okay, well, sorry about that.

Leo: So the other Steve Gibson, we will buy back that handle if you want it - if you don't want it.

Steve: Nah, I love SGgrc. I think that's established now.

Leo: Yeah, it's short, which is great, as you said.

Steve: And short is what you want. So the alternative solution is what's called a CNAME record. It's always been in DNS, thanks to the brilliance of those guys. And it's essentially, it's called an "alias." And a CNAME record says that, yes, you found the DNS for this domain name, but that's an alias for another one. And so essentially what that does is it's a DNS redirection rather than a browser protocol redirection. And that's how, up in the URL, when you're at steve.grc.com or blog.grc.com, that domain name is still in the URL. So it didn't change to WordPress.com slash something down in the directory.

Leo: Which a 301 would do; right?

Steve: Yes. And so I pay WordPress some nominal fee, like \$5 or something annually, for their support of that because they need to essentially jump somebody down into their directory hierarchy to where my blog stands in that directory hierarchy, in other words, under the AgileSynapse subdirectory, to find the blog. So there is a tiny little bit of work on their part, and there would be none if I used a 301 Redirect. But it looks more like it's a blog coming from me.

What broke? What broke was when I was talking recently about Strict Transport Security. I've had that in place now for a long time. But in covering the updates to it, as we did a couple weeks ago, I saw two new parameters that could be added. One was "include subdomains," and the other was "preload." And I thought, oh. And I thought, okay, let's see. Are there any subdomains that I have at GRC that should not be security enforced? And I thought, no, you know, media.grc, www.grc, GRC itself, I want security enforcement on those. So I added "include subdomains" to the header which the GRC server emits.

But there's a problem because what that told all the browsers was that every subdomain of GRC needs to have HTTPS enforced. And blog.grc.com and steve.grc.com, to the browser, look like domains. I mean, they are. So the CNAME comes along, and it transparently switches the browser to WordPress's domain, and the browser checks the certificate. It's looking for GRC.com, and it gets WordPress.com. And it says, huh?, and won't allow the connection.

So a couple weeks ago I broke this CNAME redirect of those two subdomains that, I mean, they don't really have to be secure. They're off in the WordPress boonies. But by doing that, and using Strict Transport Security, I made it impossible to get to those sites. So I went back in, simply removed "include subdomains." And this was really perfect because, after I did that, I went to Firefox and tried again to go to steve.grc.com, and it said no. Well, it said no because it was holding onto, as it should, the most recent instance of the Strict Transport Security header that it had received from GRC.

So I thought, okay, yeah, that's good. So then I went to GRC, just brought up the home page, refreshed it for good measure. Then I went back to steve.grc.com, and it came right up. So the Strict Transport Security header was refreshed by an actual visit to GRC, losing the "include subdomains" tag, and it all worked again.

Leo: Someday would you - I would love an explainer on what all this junk means. CNAME is canonical name. And I use it, too. As you can see, this is a DNS record for Leoville.net. And so that was - I used to host it, but now I've moved it to be hosted by WithKnown. So I had to add a CNAME there, and an A record to point to their server IP address. I have no idea what I'm doing here. I do this all by rote. So if you ever wanted to do a show where you explain what this...

Steve: On configuring DNS.

Leo: Well, what does it mean? You know, part of the problem is every domain name system has - this is Hover.com, but all the registrars have different systems. Your SoftLayer has a different system. So, but it all resolves down to something like this, this record. And I'd love to know what that stuff, all that means. It'd be a great subject.

Steve: Yeah, and actually that sort of human readable record is a proxy for the actual BIND format record, which is way less intelligible.

Leo: Oh, there's weird numbers. There's 1799 and stuff. There's all sorts of weird stuff in there.

Steve: Yup, versions and dates. And MX records, of course, have a priority on them, due to some imagined future that actually didn't occur, where you would have different - you'd have like a hierarchy of mail servers, and they would have priorities in terms of what someone wanting to send mail would try first and try second and try third and so forth.

Leo: Right, MX records and...

Steve: Or they could have the same priorities, and then it would be round robin. So, yeah, interesting.

Leo: Crazy. And I know enough to cause real problems. So I would like to know more.

Steve: It's always nice when something or someone does that for you because it's like, oh, thank god.

Leo: Well, I had - it's funny because I had conflicting hosting information in there, and it wasn't resolving. And Ben Werdmler, who's great, at WithKnown, gives great support, said we'll do an nslookup on that, and you'll see what's going on. And, like, it was - half the time it was going to Dotster.com, half the time Hover.com. It was very confused. But nslookup is a very useful tool there, too. Is there a way to dump the BIND record, like you could see what it says? Probably is...

Steve: Oh, yeah. In fact, you can do - nslookup has some parameters. There is an "any" matching for nslookup where you can say nslookup, I don't know what the exact syntax is because I haven't used it for a long time, but you can say "any." Normally you're asking for the A record, and so it defaults to A.

Leo: Right.

Steve: But if you do nslookup and do /?, you can coerce it to give you its own help for how to do that. And the "any" dump just says I want everything from the server. And actually, there's a story I did add to the notes today. I'll probably pick it up next week. And that is, I noticed that CloudFlare seems to be lobbying for eliminating the "any" option because, boy, is that an amplification attack. Because you can do it over DNS, a tiny query saying "give me all DNS records"...

Leo: [Mimicking heavy machinery]

Steve: Yes.

Leo: Actually, somebody in our chatroom says, and this might be why, nslookup is deprecated, and you're supposed to use "dig" now. And I just went "dig Leoville.net," and, boy, you get a lot of stuff. This is actually much more useful.

Steve: Yes.

Leo: Yeah. Wow.

Steve: Wow. Deep voodoo.

Leo: Deep voodoo. Someday, will you, please? I want a show.

Steve: Yeah, okay.

Leo: I trust you. You're the only person I trust to explain this stuff.

Steve: We'll make it intelligible.

Leo: Someday.

Steve: We have, I think, this and a couple more weeks, and then it's over. So I did want to encourage our faithful listeners to vote for Security Now! on the PodcastAwards.com site.

Leo: Vote early, vote often.

Steve: Vote early, vote often. I came back the next day and it said, oh, it hasn't been 24 hours. It's like, okay, fine.

Leo: So what do you mean? So they allow you to more than once, but...

Steve: They tell you to. They say, you know, come back tomorrow. It's like, why? So we see your ads again? I mean...

Leo: That's why, yeah, there you go. You just nailed it.

Steve: So I would like, I'd just love to win. And then I'll leave everybody alone for the next 10 years. But it'd be fun to win. So PodcastAwards.com. And thank you. We're at the Technology section at the very bottom. I saw a tweet from someone who said, "Dang, Steve, you and Tom Merritt are in the same category. What do I do?"

Leo: Uh-oh. Uh-oh.

Steve: And Tom responded very kindly. He says, "I voted for Steve."

Leo: Aw. I think that means you're supposed to vote for him.

Steve: I didn't. I want to win this sucker.

Leo: [Laughing] I love it. All right. So PodcastAwards.com. Vote your conscience, often.

Steve: Vote your conscience, and vote it often. Yes. Okay, Leo.

Leo: Yes.

Steve: The Apple Watch.

Leo: Yeah.

Steve: So my biggest puzzle. So I'm just loving being alive now because this, well, to the degree that I am at the moment. This is just such a fun time because I want to see what's going to happen.

Leo: Yes.

Steve: This is really - it's bizarre, and it's over the top, and it's crazy. The biggest, well, many things stand out in my mind as being like, okay, maybe. One is that the bands are such a large percentage of the watch price. It's like, holy crap. The \$549 watch, which is the one, I call that the Mama Bear. It's not Papa Bear, and it's not Baby Bear, it's Mama Bear, the one in the middle. The \$549 watch, the final price can range up to \$1049, depending upon what band you choose. What?

Leo: Oh, it's worse than that.

Steve: The band can be as much as the watch?

Leo: If you look at the gold watch, yeah, if you look at - oh, more. If you look at the gold watch, the only difference between the \$10,000 gold watch and the \$17,000 gold watch is the band.

Steve: Yes.

Leo: And so you're telling me that a leather band with a gold clasp is \$7,000? I think that's not right.

Steve: And selling you a \$10,000 gold watch with the plastic band, it's like, come on.

Leo: Nobody's going to buy that, and that's obviously the point.

Steve: Yes.

Leo: So Kevin Rose, who is, of course, you know Kevin, former Screen Saver and Digg founder, and a watch fanatic. Kevin has in later life, I guess he's made enough money now that he can afford, like, fancy wristwatches. In fact, he's created a blog called Watchville. There's a Watchville app.

Steve: And why not?

Leo: Why not? He's rich now.

Steve: I mean, and Kevin - yeah.

Leo: So this is his take on the Apple, the expensive Apple watch. This is on TechCrunch. "The gold Apple Watch is perfect for douchebags." He points out, and I think he's absolutely right, that technologically there is no difference. The watch that you buy for \$349 is exactly the same, except for a gold case, as the watch you buy for \$17,000.

Steve: Which is really interesting. It doesn't have, like, double the...

Leo: The RAM or anything. There's a sapphire - one difference is there's a - the

more expensive watches have a sapphire screen. So he says, really, in fact he quotes Anna Kendrick, the movie star, "We should be thanking Apple for launching the \$10,000 Apple Watch as a new gold standard in douchebag detection. Anybody who's wearing that watch," he says, "is proving that they just have more money than brains." Or they're really - si it's a very ostentatious way of showing off. I'm excited about - I like wearables. You know, I wear the Moto 360, the Android Wear watch.

Steve: Oh, I'm behind you buying one. So I'm glad for that.

Leo: And Apple's done an interesting psychological thing, it's technically called "anchoring," where you start with a low price that is like the price everybody pays attention to, \$349. But it's not the watch you want. You want the next one up, always.

Steve: No one wants the cheap one.

Leo: Right. So they've established - they've done two things. By setting the outer limits at \$349 and \$17,000, they've made you not feel so bad about spending the same amount for your watch as you'd spend for your iPhone, six or \$700.

Steve: But Honey, it's, like, less than my iPhone was.

Leo: Exactly. And it's not \$17,000. So anyway...

Steve: And, you know, that lens of yours is even cheaper than the high-end watch.

Leo: I know. And the lens, I'm getting something for my money there. But the watch...

Steve: Well, see, and that's just it. I'm actually not going to buy one, at least not the first one. I remember, you and I were at this point in our relationship was when we were flying to Vancouver. So we'd stopped going to Toronto to do Call For Help, and we'd switched over, and we were going to Vancouver. During one of the days that I was doing shows with you, the controversy that day hit of the Apple, the new announcement of the next iPhone with the radically lower price. And there was a lot of conversation about that, that, well, you know, the people who were early adopters, they really wanted that iPhone at that high price. So, you know, yeah, they got an arrow in their back. But they got the phone that they wanted, and Apple - because you may remember it was a precipitous drop in price, just bang. Now the iPhone costs this. It was like, whoo.

Leo: Although I have to point out that if you got an iPhone 6 Plus...

Steve: I did.

Leo: ...and maxed it out, it's over a thousand dollars.

Steve: Yes. Yes. Yes.

Leo: So it isn't exactly cheaper. Still the most expensive iPhone ever.

Steve: The other thing about it that seems so conspicuous is somewhere in a meeting with a lot of these amazing bright creative people sitting around, someone said, you know, like Tim, he said, "Attention." He said, "Okay, we're going to do a watch. It'll do all of the expected iPhone extension things. But what more can we give it? What more could a watch do?" And someone raised their hand and said, "Oh, doodles. It can do doodles."

Leo: Yeah, that's a silly one.

Steve: And it's like, okay. Good. Write that down on the whiteboard. Anybody else? I could send my heartbeat to, like, anyone else. Oh, oh, wait a minute. Write down on the doodles who also has a watch. So this stuff is inter-watch stuff, not watch to phone. And of course this is also classic network externalities, where Apple is creating features that require one at each end. And it works because, as you said on one of the shows, Lisa is now getting one, which she wasn't planning to before, so that you and she can doodle, do intimate doodles to each other.

Leo: Exactly, yeah.

Steve: Which, you know, so, yes, it works. But it has this sense of straining for a reason to exist. It's looking for a market, rather than it fulfilling an obvious need. So although, again, for somebody who is iPhone-centric, who is messaging all the time, I mean, I'm not saying I won't ever get one. But it seems unwise to get the one in the first year. And I don't need it enough that it's like I have to be an early adopter of this thing. Well, although it does sound fun.

Leo: [Exclamation]

Steve: But, okay, but...

Leo: They got you.

Steve: Like with a really good band, \$349 with a good band, it's going to be \$800.

Leo: Yeah, I'm going to get the steel with the Milanese loop band, which is \$650, I think, or \$699.

Steve: The band, the band thing is - I don't get it. And they, like, I saw some marketing that said every single link is individually machined.

Leo: Yeah, that's BS, yeah.

Steve: And hand-tooled and polished.

Leo: That's the link, that's the link band.

Steve: We're only able to do two a day.

Leo: Nine hours. They say it takes nine hours to mill.

Steve: Right.

Leo: But that's - and Kevin Rose's point is that people buy expensive wristwatches because of the craftsmanship involved. And I think that's an attempt, you see, the watch itself, there's no craftsmanship. It's stamped out by Chinese slave labor.

Steve: And it's Apple's secret aluminum. It's their alloy. They show it glistening, going down the runway.

Leo: Yeah, yeah, yeah, those movies are good.

Steve: We're sprinkling in a little bit of magnesium, and a little bit of zinc. And this is the best aluminum that's ever been.

Leo: Somebody's saying buy it without the band. I don't think you can buy it without the band. I think that you have to - or I guess you could buy the rubber band.

Steve: You could buy it with...

Leo: And then you can snip it off and put a...

Steve: It's a rubber band. The rubber band.

Leo: I'm sorry, fluoroelastomer. They don't even want to say "rubber." But I believe fluoroelastomer is rubber. I could be wrong. Rubber band. Even - it's so funny because I think Christy Turlington, when she came onstage, the model at the Apple event wasn't briefed fully because she called it a rubber band. And I think that was a...

Steve: Whoopsie.

Leo: Who.

Steve: We don't call it that.

Leo: Who. It's fluoroelastomer, please. Now, Apple's great at marketing. There is something here. I think one of the clues to why there's drawings and heartbeats is the original Apple Watch was, according to The Wall Street Journal and a leak, supposed to be much more of a health device. But they, for a variety of reasons - for technical reasons, battery reasons, and FDA reasons - they couldn't make it the health device they wanted to. So at that point, not so long ago, they were casting around for, oh, crap. It can't do that. What else could we have it do?

Steve: Right. So it has an accelerometer and a heart rate sensor.

Leo: Which they all do. This does.

Steve: Yeah.

Leo: And that's what you can easily do.

Steve: My shoe does.

Leo: Yeah. It's not a hard thing to do. Yeah, your Nike Pod, Fuel, whatever it is, Nike Pod is doing that. So, but I do think that there is something about wearables. I figured out what it is, actually. One of the reasons I love Android is because of the widgets. I can turn on the screen, and without diving into an app, instead of, on the iPhone you see a grid of apps, and you can't really get much information from the screen.

Steve: Correct.

Leo: You have to dive into the apps to get anything out of it. But on the Android I have widgets. These are all informational. There's calendars. There's news. There's

weather. Without doing anything. And I realized Apple's never done that.

Steve: Windows has that, too; right?

Leo: Yeah, Windows Phone has more information on the screen. Apple's never done that. Apple's effectively a grid of icons.

[Crosstalk]

Steve: Yeah, well, we could [indiscernible] time, yes.

Leo: The watch is the widgets. The other thing that became apparent that wasn't completely apparent originally is that you don't - the watch does everything the phone does, just on a little screen. So you can even be Dick Tracy and talk - I have a phone call. So, and because it supports WiFi, your phone doesn't have to be within Bluetooth range, it just has to be on your network.

Steve: Yeah, that's brilliant, that it's able to use WiFi to bridge [indiscernible].

[Crosstalk]

Leo: So that's valuable to me. I can leave my phone here and wander around the studio and [indiscernible].

[Crosstalk]

Steve: It's valuable [indiscernible] everybody.

Leo: Yeah.

Steve: Yeah.

Leo: So there's some - they've found some value in it. It's not that it's completely useless.

Steve: If you were to lift your arm very slowly and kind of, like, trick it so that it didn't know you were lifting your arm, can you see the time?

Leo: No.

Steve: Or do you have to give it, like you have to do a Heil Hitler in order to get this thing to wake up and...

Leo: There is a low power mode that just shows the time.

Steve: Okay.

Leo: And I think it goes into that if you get down to a small - I think.

Steve: That display is always on? Or never on?

Leo: That one is always on, but only when it's running out of juice. Normally, it's much like the Android Wear watch, where you don't want to leave it always on. It'll kill the battery. But a gentle twist of the wrist turns it on. Or a tap of the screen. They have to do that because the problem is these devices are so small they can't hold much lithium ion. Where you going to put it?

Steve: Yeah, in the strap-on optional battery pack, you know, that goes on your forearm. Really takes the bling out of a \$10,000 watch.

Leo: This is a problem with all portable devices, mobile devices; right? You can't get too small because you have no battery life. We've got to get this - what happened to supercapacitors? We've got to get this battery life thing solved.

Steve: It'll be good. And that would be perfect for this application because a supercapacitor could charge so fast that you'd just briefly touch it to a dock. And, you know, they could set them up next to electric car charging ports and things, so that they're all...

Leo: Supercapacitor doesn't give you more capacity, it's just faster to charge it. Do you get more capacity per square inch?

Steve: It is, well, of course it doesn't do anything today because they don't really exist in a practical fashion.

Leo: Except in my screwdriver is the only place.

Steve: But if it could reach - yes. If it could reach parity with chemical storage, electrochemical storage, then you get instant charging.

Leo: Which is great.

Steve: Which is super great, yeah.

Leo: Providing enough voltage in the charger.

Steve: As long as it doesn't, like, blow up and fuse and turn into a...

Leo: On your wrist.

Steve: ...a microwave on your wrist. Yeah, you don't want to burn, don't want a \$10,000 shaped spot on your wrist when this thing goes nova.

Leo: I get some value, not enough to make it a must-have, out of my Android Wear watch. I anticipate a similar kind of reaction to the Apple Watch. Of course I will buy one, and I'm not going to buy the anchor watch because that's cheap.

Steve: That's right. It's cheap.

Leo: So I'm going to buy the middle one, like everybody else. Which is ridiculous. And I can tell you this. If I didn't have this job, I certainly would not. But I'm also buying a MacBook. What do you think of the new MacBook?

Steve: Oh. Oh, yeah. I'm - it's just...

Leo: You crack me up.

Steve: It's Apple at their finest. I love the...

Leo: Steve, there's a lot of people in the chatroom are saying, oh, no, this thing is a Netbook for double the price. Because the Core M isn't a superfast processor. But, you know what, processors are pretty fast nowadays.

Steve: That's right.

Leo: Why are we turning up our nose at 1.2 GHz, really?

Steve: I never liked that, like, basically they fixed everything that annoyed me. I never liked that weird, like, front of the pad click-down thing. That seemed wrong.

Leo: Yeah, this is, I think, the Force Touch is a great idea.

Steve: Oh, it's beautiful, yes.

Leo: It's thin. It's light. Two pounds, which is really pretty light.

Steve: Oh, lord.

Leo: Are you worried about one connector, though, the USB FC?

Steve: I guess I'm worried about one odd connector because I don't have any of those USB devices. Maybe we'll start getting them.

Leo: Yeah, Google's already announced a new Pixel that uses two of them. Apple should have done two, by the way.

Steve: Yeah, I think so.

Leo: Then it would be easy; you know? But they only did one.

Steve: But my point is that we will then need an outboard converter. And I've always been a little annoyed by Apple, like the main device sells at a good price, but all of their little accessories really do seem jacked-up profit margins for the other thing that you need because, oh, your TV doesn't have a lightning connector? Well, you're going to have to get [indiscernible].

[Crosstalk]

Leo: Well, here's the good news. Type C is not, is no longer proprietary.

Steve: Right, right.

Leo: In fact, that's kind of stunning. I can't remember Apple ever - or most laptop manufacturers ever doing nonproprietary power connectors. That's a profit center. So that in my opinion is a good thing. And by the way, USB Type C, because I had to look this up, not only does it power, but it can power up to a hundred watts. You can put a lot of power across it.

Steve: Hundred watts.

Leo: I know. I don't think any laptop does that.

Steve: A lot of those pins must be conveying power. And so you can use as many as you

like.

Leo: I think you're right. I think it's probably like that. If you're going to use it for video, you obviously aren't going to put a hundred watts of power across it.

Steve: Exactly.

Leo: Yeah. I'm really - I think that, because Apple will jumpstart USB Type C, we're going to see it everywhere. It is a standard. And that is a good - if the one thing accomplished by the MacBook is that laptops, like phones, have a single standard adapter for their power, that's huge. And at least the dongle you buy is going to be usable with other things. It's not just an Apple dongle.

Steve: And it also means, yes, it also means crazy video. I mean, through - that means that USB-C is now fast enough.

Leo: Yeah. There's two 10-gigabit channels on it.

Steve: Oh. Oh, 10 gigabits.

Leo: Two.

Steve: Oh, nice.

Leo: Yeah. So that's good. I'm excited.

Steve: Yeah.

Leo: If for that alone.

Steve: I just love gorgeous engineering. And the new MacBook Air is just - it is just sex. It is, you know, the idea that they say, okay, well, we want a wedge-y sort of overall profile. So that means that we're going to have a wedge-y sort of interior cavity size because the screen is going to be flat. But if we have a standard prismatic cell, as they're called, the lithium polymer cell, which can be any shape you want as long as it's square, then we're going to lose a lot of space. And in their video they just show that so elegantly, where at any thickness, this thing can't, like, come down into the nose, into the front edge of the laptop because it's too thick.

So Apple scallops their batteries in a series of multiple scallops. They figured out how to come up with a layered battery that ends up being one unit, but where, as it goes lower, these scallop sizes pull back, and the effect ends up being the equivalent of a wedge-shaped battery that is then able to tuck itself right down into the smallest interstices of

this case. And I don't remember what the number was. But it resulted in substantially better overall power delivery.

So I just - and they took the existing logic board that was already high density and said, okay, this is three times bigger than we need it to be, or maybe it was four. We're going to cut this thing down to much smaller. And when you look at it, it's like, oh, my god. I mean, so they designed it so that the components would nest into each other. And of course the big change is no moving parts, no blower any longer. So they were also able to remove the fan completely. So not only is it always silent, but boy, getting rid of that is a great achievement. So, oh, I think that is - I think currently it's the penultimate statement in a fully functional portable laptop computer.

Leo: But you use the iPad for, like, you'll take the iPad to Starbucks; right? I mean...

Steve: Oh, my lord, yes.

Leo: So this is kind of midway between a MacBook Pro and an iPad. It's kind of - it's light, it's thin. By the way, not the lightest and thinnest notebook out there.

Steve: Oh, and the new design of the keyboard, so that the keys are not kind of rocky tilt-y any longer. They came up with what they called a "butterfly" design, so that they were able to get even less height, yet like a firmer feeling key, so that you can still type on it. And the keyboard also goes edge to edge on the - anyway, I don't mean to sound like an Apple fanboy on this. But, boy, you know, I have...

Leo: Well, those keys are not going to have the travel of previous laptop keyboards, so...

Steve: They can't.

Leo: They can't. There's nowhere for them to go.

Steve: There's nowhere for them to go, yeah.

Leo: So I've used, like, my ASUS, I'm sorry, my Acer S7, which is equally thin, the keyboard was almost unusable at first because it had so little travel. I made a lot of mistakes on it. They improved it in the second generation, the Haswell version. So I'm curious. We'll see. You kind of got used to the S7.

Steve: At some point it becomes too much of a compromise.

Leo: Right. Haptic, I think haptic on the trackpad, though, I can't wait to see what that feels like.

Steve: Oh, yes. And on the Watch. The Watch has it, too.

Leo: Rene Ritchie, who was there - and the Watch has it, too. Rene Ritchie, who was there, and Serenity Caldwell and Jason Snell, all said it's kind of uncanny, even though the trackpad doesn't click anymore, because of the haptics, it feels - you feel like there's a physical thing happening. It tricks you. And so I'm curious about that.

Steve: Yeah. And then of course the deep touch, or power touch, or power click, or...

Leo: Force Touch.

Steve: Force Touch, right.

Leo: Which is actually a terrible name.

Steve: I know, I thought the same thing.

Leo: It's not a good name.

Steve: While Tim was saying it, I was thinking, well, how about something else?

Leo: Yeah. Force implies that you're using force. And I would hope it's not like I have to [groan] push down on that thing [groan]. More touch, slightly more touch would be good.

Steve: I think they've just done a great job.

Leo: Well, they've obviously won you over. I presume you're going to order one immediately.

Steve: Yeah, I like my existing Air, and this one replaced it. It's just, as I said, it's just a work of art, a work of engineering art.

Leo: I'm going to get the gold one because I'm that douchebag.

Steve: I have a gold S, I mean, a gold iPhone 6 Plus.

Leo: It's kind of - it's not like bright gold. It's kind of a nice, tasteful...

Steve: And we know it's not real gold. It's just painted on.

Leo: Yeah, it's gold paint.

Steve: So it's like, you know, yeah, gold tone.

Leo: At least they're not charging a hundred bucks more for the gold one.

Steve: My god, or \$10,000. So my only mention this week of SpinRite was just - it's from a tweet that came in on the 11th, which was - what is that? Yesterday, at 7:22 in the morning, Michael Parker tweeted: "@SGgrc Hey. A decade of waiting to use SpinRite, but never found a need. My day finally came. It saved my SSD, which would not boot. Thanks."

Leo: Aw.

Steve: So further substantiation of what everyone is seeing, which is SpinRite has a bright future because it moves right into solid state mass storage without skipping a beat. So thanks, Michael. I did thank him via the Twitter for his note and for implicitly allowing me to share that with our listeners.

Leo: Steve Gibson, you are a trooper. We hoped you'd be better by now, but a cold is a seven-day affair, isn't it.

Steve: We're going to get it done.

Leo: Thank you for being here.

Steve: Glad to be.

Leo: If you're tuning in right now, we've got about a half an hour before TNT, and we're going to get the FREAK and the RowHammer on right now.

Steve: Okay. So this is bizarre, FREAK. I got a lot of tweets from people when the news broke, saying, oh, my god, should I worry about this? What should I do? And the answer was it's a bizarre thing that was discovered in today's contemporary, as of two weeks ago, SSL/TLS secure network stack software, existing in Apple's own SSL, their secure transport technology, and the most recent version of OpenSSL, actually before v101k, are vulnerable. So, and the OpenSSL guys were given a heads-up about it last month.

But it's not something that the end user has to worry about. And unfortunately, this was another one of those things where the popular press jumped on a sort of true but technical aspect that sounded really bad. What was discovered was a surprising cipher

suite downgrade attack that would - and this is what's shocking - that would downgrade to a cipher suite that nobody even supported. It's like, what? What?

Leo: Well, there's a reason nobody supported it.

Steve: Well, but, okay, but there's a subtlety there, too. The way TLS works is that your client gives the server a list of cipher suites that it knows. And then the server looks at its list and picks the one that is the best, that it considers best, from its list that the client also supports. And then it uses that to establish the cryptographic connection. Long, long ago, as you said, Leo, and this ties back to what I was talking about before, the export-grade cipher suites were eliminated. So, and the way that actually feels is that, in a list of cipher suites, we've talked about GRC's list, I think it's - I have a bit.ly link, bit.ly/grcciphers. And that gives you a text file of the list of cipher suites. And among those are RC4 at 40 bits. And it was like, you know, and 40 bits is the symmetric key length. And with that comes RSA-512, which is 512-bit for the public key component of the cipher suite.

Remember that we both need a cipher suite, and this is why it's called a "suite," is it specifies it's a chain of different crypto. There's the specific public key negotiating side, the shared symmetric key technology and length, and then how they do message authentication. And so a given one of these specifies those things. And I saw them a couple months ago there in the list of ciphers that Windows Server 2008 R2 that I'm using offers. And they're like, they were down at the very bottom of the list. And I only had, what, 2K of registry space to list mine. So any sane person would never get down that far. But you'd also never choose that. You would never, like, say, oh, yeah, I want to, today, in 2015, I want to support a 512-bit RSA.

Now, we're still using 1024-bit RSA and moving to 2048. And it's important to remember that 512 is not half of 1024. It's half in terms of bit length, but it's 2^{512} . The 1024, RSA-1024, while only being twice as long, is 2^{512} stronger. So it's a radical jump from 512 to 1024. And we're making another similar, even more radical jump when we go to 2048. That's 2^{1024} stronger than 1024-bit RSA. So what we're talking about is a dawn of the Internet, like the original Netscape browser's SSL, when strong crypto was still regarded as a munition, therefore in this weird case you could use it in the U.S., but not outside of the U.S. So browsers offered their list, their sort of a curated list of cipher suites that are strong, that they support. The server does the same thing.

Here's what some researchers discovered completely out of the blue. Remember that cipher suites all come with a commonly recognized hex designator. That is, all of them, there is that long textual form, but there's also just a simple two-byte designator to say - sort of give it a family and within the family designation. And some standards body, IETF or ICANN or somebody, probably IETF, have standardized on what these numbers are. What these researchers discovered, and they stumbled upon this because they were - they wanted to experiment with building a very strong fuzzer for SSL and TLS. A "fuzzer," of course, is the industry's term for just throwing all kinds of random stuff at an API and see if you discover any surprising breakage, something that it's not supposed to do.

And here's what they found. If a man in the middle intercepted the client's opening handshake to the server, and that opening handshake is going to contain a range of cipher suites that we would all think in this day and age are strong, even Chrome, for example, we know Google would be the last place on Earth that any browser is going to send a weak cipher suite. So the man in the middle intercepts that, removes them all,

and puts in that designator for export-grade crypto from the dark ages and forwards it on to the server. The server, 37.6% of the servers on the Internet will go, oh.

Leo: Yeah, sure. You're stupid, but I'll be stupid, okay.

Steve: Shoot. If that's all we can do, then I guess so.

Leo: Wow.

Steve: I mean, there is a cipher suite that is no cipher. There's like a null. I think it's zero. And fortunately I think they don't respond to that. But the 36.7%, or 37.6, something like that, more than a third, said, oh, okay. And so the server then responds to the client, and the client, who never even offered that cipher suite, but sees that the server has decided to use this export-grade cipher, says, hmm, well, I have that. So, okay. So this begets so many questions. First of all, why are these even in any OS at all? There's absolutely no reason for it. The null cipher is gone. Kill this one, too.

But the point is that no one did. OpenSSL was vulnerable. OpenSSL, you might argue that because it has such a wide range of applications, and many of them are like it's the development and testing platform for TLS, they're going to keep them all around. Okay. So maybe allow them to be installed, or allow there to be some sort of special access to these really, really brain-dead ciphers. But don't advertise a secure set and then accept one that you never advertised.

So that was what they found. They found both that a shocking number, I mean, even Apple's Secure Transport was vulnerable to this. A shocking number of existing systems - oh, and IE, that's what Microsoft fixed on March 9th was they removed this "characteristic" from Internet Explorer. It, too, was vulnerable to this. So not only were these absolutely no longer can we even consider this to be secure cipher from the OS, but it's still there. It also turns out that you could just use one of these that was not advertised, and both ends would say, oh, I didn't really think you were going to come back with that one, but okay, fine.

And so that's FREAK. The response, as you can imagine, has been swift, as swift as possible. Apple's fixed it; Microsoft's fixed it; OpenSSL has had it for a while. Akamai, I had in my notes here the number of - okay. So who had vulnerable servers? 36.7% of the 14 million sites serving browser-trusted certs, meaning that in a scan on port 443, you don't know for sure that you're probing a website because you're not scanning by domain name, you're scanning by IP address. So you're just covering the entire Internet. But when it returns a cert that browsers trust, it's a very good bet that this is a web server.

So that was their selection criteria. So the Internet had 14 million sites serving browser-trusted certs, and of those 36.7% were vulnerable. But many of those, it turns out to be Akamai content delivery network, CDN, endpoints. So that skewed the numbers a little bit. And Akamai is busy, they're going to be doing like a major retrofit in order to bring their whole network up.

But there were some embarrassments in here, too. The U.S. government sites, such as www.nsa.gov...

Leo: What? What?

Steve: ...was vulnerable to this; www.whitehouse.gov and irs.gov. And the FBI tips site, tips.fbi.gov, all vulnerable. And the Facebook.net site, it's connect.facebook.net, that's the domain name that serves the ubiquitous across the Internet "Like" buttons from Facebook. It was vulnerable. And an attack, a man-in-the-middle attack on that could have had huge consequences for the security of Like buttons everywhere. So this thing was - it was pervasive. In the show notes, if anyone's interested, I have a complete list of the vulnerable TLS client libraries and the web browsers.

Well, in fact, so Chrome, versions before 41 on various platforms were vulnerable. So I just said that Chrome wouldn't be advertising insecure cipher suites, but even Chrome left them in there for, like, no explicable reason. Internet Explorer on OS versions before March 9 are vulnerable. Safari on OS versions before March 9 are vulnerable. Opera, versions before 28, which I'm sure is recent, are vulnerable. Android browser is vulnerable, switch to Chrome 41. And, you know, we've discussed before how unfortunately Google, for just practical reasons, is unable to go back in time and service the millions of very earlier version Android browsers. They're always going to be vulnerable.

The good news is the sites they are probably going to be connecting to will not be vulnerable. And you need this vulnerability on each end. So essentially this won't affect Android users even of early browsers because there'll be nobody vulnerable for them to connect to. And the BlackBerry browser is vulnerable, and we're still waiting for a patch for that. I'm sure BlackBerry, because they're a going concern, will fix it.

Leo: Sort of a going concern.

Steve: Well, yeah, exactly. At least, you know, still interested in...

Leo: Still in business, yeah.

Steve: ...not having any overt security flaws in their products.

Leo: No, in fact, that's a big part of their product pitch, isn't it.

Steve: Yeah, exactly. So really interesting discovery. It's like something that has been lurking in our TLS libraries from the beginning that we just didn't see.

Leo: Did you check if ClintonEmail.com was vulnerable?

Steve: I had that thought, actually. I didn't see that Windows Server was ever vulnerable. And ClintonEmail.com was using Windows Server 2008 R2...

Leo: Yes.

Steve: ...as its OS platform.

Leo: Although they didn't get secure certificates for the first three months of her term as Secretary of State, so...

Steve: Wow.

Leo: Yeah, that's just come out. So, but it is secure now, if you'd like to use it.

Steve: Well, yeah, exactly, yeah. Okay. Our final topic.

Leo: RowHammer.

Steve: RowHammer.

Leo: Like this name.

Steve: Exploiting DRAM. Exploiting DRAM. So this began with a team of researchers at Carnegie Mellon University, which I in the past have referred to as Carnegie Melligan, but we don't do that anymore. Carnegie Mellon.

Leo: Really? Carnegie Melligan?

Steve: I said Carnegie Melligan once, and it's like, okay, what?

Leo: That's the Irish, the Irish - it's in Dublin, Carnegie Melligan.

Steve: It has sort of more of a rhythm to it.

Leo: Yeah, it does, it's nice. Actually, I want to start a university called Carnegie Melligan.

Steve: Yeah.

Leo: I like it.

Steve: So these researchers did the groundbreaking research. And in the show notes is a link to a 12-page PDF that is really interesting, has a lot more details than I'm going to go into because I'm going to go into plenty as it is. They discovered a way of affecting, deliberately affecting bits in RAM that were not otherwise under their control. That is, when a process is in an operating system, it has RAM that's been allocated to it by the OS. And so, but part of the containment that the OS provides through the page table mechanism is this interprocess isolation. That is, a process sees RAM, and it might even see the same address ranges of RAM, as some other process. But the page table mechanism in contemporary CPUs, pretty much all of them that support virtual memory, virtual memory is the name for this, is such that when the process generates a fetch of any kind, or a write, that address is mapped through several layers of indirection.

So basically what looks like an actual memory address is actually a pointer to a slot in a table that the processor uses to map that to a physical RAM address. And this becomes important in a minute because of the work that the Code Zero guys at Google followed up this initial groundbreaking research with. So what these guys at Carnegie Mellon discovered was that they could perturb bits in RAM in adjacent rows, thus the term "RowHammer." We've talked about DRAM before. DRAM is the only technology which has steadfastly resisted the normal performance increase curve that everything else has gone through. It used to be about the same speed of the processor, but the processors just went crazy. And the speed of RAM, while increasing, really lagged behind.

At the same time, the density of DRAM has increased so that we've got an incredible amount of DRAM now on a small set of chips that was never possible once upon a time. The Intel, the first DRAM was I think the 11.03? Maybe it was 11.02, but it was 11.0 something. And I think - and it was like the classic DRAM. It was 1-Kbits. And so that was addressable as 10 bits. So that would be five bits for row and five bits for column. So that's a 32 by 32 array of bits, which was all we could do back then. And everyone thought, oh, my god, that's - we don't have any, so thanks very much, Intel. Now we have gigabits. We have that by dramatically shrinking the cell size of DRAM. And DRAM is essentially capacitors, which can be charged up to a certain level. And just through the nature of how small they have become, they tend to discharge by themselves.

So the idea with DRAM, it's really nice that we've got this, like, this incredibly low-component-count cell. But because it isn't very static, it's necessary to come back around and read it before it has had a chance to discharge into the uncertain region, where we're not sure if it was ever a one or it was always a zero. So thus DRAM must be refreshed. And it's typically refreshed every 64 milliseconds. What these guys discovered is that, if they could arrange to hammer the adjacent row in DRAM a huge number of times in between refresh cycles, they could flip some bits.

And they didn't do the work of exploiting it. But to their credit, the Code Zero team did. Google's Code Zero team picked up this potential problem and basically weaponized it. They have two proof-of-concepts now. One was an attack on their own Native Client technology in Chrome. Native Client is the very impressive but kind of scary idea of running native x86 code in the browser. Rather than JavaScript, you run actual machine language, thus giving your browser the same performance on a browser-based application as a native application on the computer would have. To pull this off, they do all kinds of voodoo, among which is they break the instructions into 32-byte blocks and analyze them in those pieces and make sure that no skewed analysis could create a privileged instruction.

So what they do is they only permit a subset of x86 instructions that are safe to run in sort of this scary sandbox that they create. But it turns out that, if you were to flip a bit in one of these 32 blocks of machine code, you could turn what was previously scanned

and checked out as being safe into being unsafe. And so they created an exploit that actually managed to break out of their own Native Client sandbox. And it was the guy that was the project lead on this, he had been the developer and the previous cracker of the Chrome sandbox, so he chose that as his first target.

The second proof-of-concept exploit they pulled off was against a 64-bit Linux system. And what's scary is it is shockingly powerful, and we don't have a solution for it. The good news is it's because Linux broadcasts a processor's page table, called the "page map," that the processor can see how its physical memory is mapped into its logical memory. That's sort of the - it's necessary for this particular exploit. So, for example, I'm not aware of any way for a non-driver in Windows to access that mapping between virtual and physical memory. But Linux makes it simple. It's `/proc/PID/pagemap`, and it gives it to you. So by using a short loop in machine language, they were able to flip a bit in their own page table which gave them global access to the system's memory. They were able basically to take control of their own process page table from the OS, giving them complete global access to the system's RAM. And this is, I mean, just an amazing work of engineering.

What I did want to share, just in closing, is that they created a GitHub site where they have a self-test for Linux users. And I've got it in the show notes. It's in GitHub, `/google/rowhammer-test`. And in Google's testing, half of the 30 laptops they tested, they were able to exploit. And in the Carnegie Mellon research, almost all of the DRAMs, they had 129, I think it was, different DRAM modules from three major manufacturers. They were able to compromise 111 out of the 129. And they noted that the problems seemed to be recent. It was new models of DRAM that was having the problems as opposed to old. So it's looking like the cell size has recently shrunk in order to bring up the density to such a point that we're seeing the integrity of DRAM damaged to a point that it's now exploitable. So that's RowHammer. I'll probably have a few more words about it next week. But I wanted to let everybody know what it was and what was going on with it.

Leo: Thank you, as always. You're the best, even on a day when you're feeling not so hot. And thank you for making that effort. Hey, one note I just wanted to pass on. This came in as we were talking, and I know many of our viewers are science fiction fans. Terry Pratchett, who is a wonderful, wonderful author, he created the Discworld series, they're really comedy-fantasy novels, but just brilliant, he's passed away. He was suffering from early-onset Alzheimer's. He called it the "embuggerance" and actually continued to write with some effort. His last novel came out last year.

Steve: And also to reread what he'd just written.

Leo: Now, come on.

Steve: Oh, okay, sorry.

Leo: But you're right. He certainly joked about it. But it's very sad because he was only 66. He was one of the, I think, one of the great minds, one of the great writers. If you have not - this is an opportunity, if you've not read the Discworld series, to

read some of it. But very sad, Sir Terry Pratchett passed away at the age of 66. And this would be a good time to break out those Discworld books. There's something like 70 of them.

Steve: Oh, my goodness.

Leo: Yeah.

Steve: All by him, or by..

Leo: Yeah, yeah. The last one, he had a device he was putting on his head to kind of help him because it was hard to write, of course, towards the end. But he managed to come out with one last one last summer. And I haven't read it.

Steve: I think actually the problem was probably that he wrote 70 of those novels, and it damaged him.

Leo: I don't know. I don't know. It's very sad. He had suffered from, I mean, 66 is very young to pass away from Alzheimer's. He contracted it I think seven years ago, so he was in his late 50s when he contracted it. So for those of us in our late 50s, that seems too young. Anyway, a great loss. But read his stuff because it's just a great - and it's all on Audible, if you want to listen. I just love it.

Steve Gibson is with us still. Despite a cold, he is not, in fact, incapacitated, as you can see. His mind works under all circumstances. You can find him at...

Steve: The show must go on.

Leo: It's amazing that after, and we've been doing it for 10 years, that this is the first time you've missed an episode. And you didn't miss it, but first time we've delayed an episode due to illness. So thank you for coming in early.

Steve: So Q&A next week.

Leo: Next week Q&A. So go to GRC.com/feedback to leave your questions. Or tweet him, because we often take tweets in the questions, @SGgrc. You'll also find SpinRite at GRC.com, the world's best hard drive maintenance and recovery utility, and lots of freebies. And 16Kb audio versions of this show, which sound kind of like they were recorded by Thomas Edison in his lab, but they're small. Smallest version, though, is the text version. Elaine Farris does a great text transcription of each and every episode. Those are all at GRC.com. We have high-quality audio and video versions at our site, TWiT.tv/sn, and wherever podcasts are found. And I hope you'll subscribe so you don't miss an episode. This is one of those shows, almost 500 in

now, where you really - 498.

Steve: Two more, baby.

Leo: And, you know, TWiT just did its 500th Sunday. So this whole network is starting to show signs of aging. But we soldier on. Thank you, Steve.

Steve: Thank you, my friend. Talk to you next week.

Copyright (c) 2014 by Steve Gibson and Leo Laporte. SOME RIGHTS RESERVED

This work is licensed for the good of the Internet Community under the Creative Commons License v2.5. See the following Web page for details:
<http://creativecommons.org/licenses/by-nc-sa/2.5/>