SECURITY NOW!

Transcript of Episode #588

Listener Feedback #243

Description: A wonderful quote about random numbers, our standard interesting mix of security do's and don'ts, new exploits (WordPress dodged a big bullet!), planned changes, tips and tricks, things to patch, a new puzzle/game discovery, some other fun miscellany - and, finally, 10 comments, thoughts, and questions from our terrific listeners!

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-588.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-588-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We've got a lot of security news to go through, including an homage to the Network Time Protocol. What other show are you going to hear that on? Well, maybe FLOSS Weekly. And questions and answers. We haven't done it in a while. We've got some great ones. Stay tuned. Security Now! is next.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 588, recorded Tuesday, November 29th, 2016: Your questions, Steve's answers, #243.

It's time for Security Now!. Here he is, ladies and gentlemen, the man of the hour, actually more like the man of the two hours, Steve Gibson, GRC.com.

Steve Gibson: Ah, but who's counting?

Leo: Yeah, no, no one. No one. More is better when it comes to Security Now!.

Steve: More, more, more. Yeah, people say, could you, you know, if you have too much news to talk about in one podcast, how about doing two a week? And it's like, okay, wait a minute. This is already, like, 18, well, no, it's more than 24 hours because I start the day before. I run through, over lunch for a few hours, I go through my entire Twitter backlog of a week in order to catch up, and then dump the mailbag and go through that and pull everything together. Basically it's a full day out of my week that I invest in assembling this. So, yeah, I'm not doing this, I can't double this pace.

Leo: Enough, enough.

Steve: I'm already...

Leo: Yeah.

Steve: But the good news is, after many weeks of trying, we have achieved a Q&A.

Leo: Wow. Who woulda thunk it?

Steve: Without skipping anything important, it was just sort of a little - maybe it was Thanksgiving. I did ask last week, please, you know, just eat your turkey, you hackers, and leave us alone.

Leo: Maybe they did.

Steve: And maybe they did, took a few days off. But we have a wonderful quote about random numbers that just tickles me. And then our standard interesting mix of security do's and don'ts; some new exploits. In one case WordPress dodged a big bullet, and our listeners are going to get a big kick out of the mistake they made. And the details of it, I think, are really interesting. And then there are some planned changes; some tips and tricks; some things to patch. I have a new puzzle/game discovery which is free. Unfortunately, it's iOS only, and I prefer things that everyone can have access to. But really an interesting twist. And we have a little bit of miscellany. And then, as many as we can get to, I've got 10 lined up here, questions and comments and thoughts from our listeners. So I think another great couple hours.

We had a story, actually it was a SpinRite testimonial. Someone was using SpinRite to recover hard drives for people. And in some cases they said, I mean, he would say, okay, it looks fine. And they go, well, you know, I needed a bigger one anyway, or I want to get a new one. So he would recover their data, clone it to a new drive, and then wipe their drive. And he said, because he's also a Drobo user, he said, "If the drive that I got from them, which they were retiring, even after SpinRite fixed it, if it was larger than the smallest of my Drobo drives, I'd just pull out that Drobo drive that's the smallest and slide that larger one in." And so as he was working on people's machines, over time drive sizes tended to inflate, as we know, and so his Drobo kept pace, just growing over time, always replacing the smaller with a larger.

Anyway, this week's wonderful quote, I just love this, this is Robert Coveyou gets the credit for this. And I just - it's just so charming. He was quoted saying: "The generation of random numbers is too important to be left to chance."

Leo: Very funny. I like it.

Steve: Love that. And our picture of the week is a wonderful Venn diagram. Thanks again to some Twitter follower who found this and said, "Oh, I've got to make sure Steve sees this." So for our listeners who can't see this, a Venn diagram, of course, is the famous overlapping circles; right? So we have three overlapping circles. So they have regions where there are only two overlaps. And then of course in the center is where they all participate. And this is really clever. And this, of course, we've talked about exactly this property before. So the circles are labeled Secure, Fast, and Cheap. And of course these are competing problems. We would all like something to be secure, fast, and cheap. But of course that would be the center where they all three overlap, which in this diagram is labeled "Not gonna happen."

But then you have, of course, three other choices where you've given up one of them. So, for example, if you just have secure and fast, okay, and you've just given up on cheap, then it says in the picture, "This is going to cost you dearly," to have secure and fast. Or get rid of speed. Get rid of fast, so just secure and cheap, and that says, "It will be ready right after you need it." And then, finally, with cheap and fast, okay, and for some reason there's no security involved, it just says "Good luck with that." So anyway, sort of a distillation of principles we've been discussing for years in a nice little Venn diagram.

And I did find among my tweets, from someone who - he's in Germany, and he signed his note "Clemens," but his Twitter handle is @AvocadoDiaboli. Anyway, he said...

Leo: He's Devilcado.

Steve: I think he is. He said: "Hi, Steve. Thank you for your coverage of the OAuth analysis in SN-585. I am friends with two of the paper's authors, and they're very grateful that you presented their work. One of them said about you: 'Finally, someone understood our paper. Steve might have understood it even better than some of the paper's peer reviewers.'" So, he said, "Thanks, and keep up the great work. Greetings from Germany, Clemens." And so it's nice, you know, we're getting that a little bit more now as we tackle some of these tricky security whitepapers because I like to figure out what is actually going on and explain that to people. So it's fun to hear back from the authors. It's like, hey, thank you. You actually understood that.

Okay. So, Leo, I know you know about this. And I just love the story title, which actually appeared on the screen, which is, "You hacked. All data encrypted."

Leo: I know where you're going with this one, yeah.

Steve: Uh-huh.

Leo: It's just really stunning, frankly.

Steve: Free rides for everybody last Saturday. So that was the message on the San Francisco Muni, as it's called, the municipal transportation system, the computer access screens across the city; which, as a consequence of this hack, which actually was apparently a cryptoware attack, gave passengers free rides all day Saturday because the entire fare processing system was down in a cryptomalware cyber attack which reportedly knocked out 2,112 computer systems citywide.

The local CBS affiliate in San Francisco reported that their inside sources said that the system had been hacked for days, and the equivalent of $73,000 in bitcoin was the requested ransom. And along with that came an email address that was a well-known email address to the people behind this particular flavor of cryptomalware, so it all looked legitimate. And so that email address was used to contact the hackers, that is, the security industry wanted to say, you know, do you have any comments? And so clearly English is not their first language. Their reply was: "We don't attention to interview and propagate news."

Leo: What?

Steve: Yeah. "Our software working completely automatically, and we don't have targeted attack to anywhere," meaning this was just a gift from some poor muni employee that clicked on the wrong link. And then they continue: "SFMTA" - which is the San Francisco Municipal Transport Agency - "SFMTA network was very open, and 2,000 server/PC infected by software! So we are waiting for contact any responsible person in SFMTA. But I think they don't want deal. So we close this email tomorrow."

Leo: Wow.

Steve: So the SFMTA officially confirmed the hack, which, you know, it's like, all the gates were up. And it's like, gee, are you guys having a little problem with your fare processing system? So, and then they said, so they confirmed the hack - I love this - but said it has not affected any service, and they refused to provide details using the excuse of an ongoing investigation. You know, we're not going to provide anymore details due to the ongoing investigation. Of course, meanwhile, the city's Metro gates were wide open because they weren't. I mean, they had to not have service affected, but people couldn't use their passes or get tickets at all, so everything was just wide open.

Leo: You can't blame them for not, I mean, I think it's reasonable, at least initially, to not say anything publicly - Muni I'm talking about - because you don't want to give any attacker any surface. But ultimately I'd be very curious as to what happened. It sounds like the fare transaction computers were affected. They have many systems, obviously. And it's apparently not interlocking, thank goodness, because the ability to run the system, which is all computerized, continued.

Steve: Yeah. Sources at the SFMTA did say that the hack affected employees.

Leo: Oh, that's too bad.

Steve: And that they were not sure they would get paid this week. And I put in parens, "(Happy holidays)."

Leo: Wow.

Steve: So anyway, and also the attack hit the Muni's email systems. So there was - so you're right, Leo. The critical transport infrastructure, like Bart, the computers that run Bart, for example, they...

Leo: You don't want those to be down.

Steve: Exactly.

Leo: Because then you can't run the trains at all.

Steve: Right. But so it sounds more like the management side, the employee interface side is how [crosstalk].

Leo: They mentioned, I seem to remember they mentioned QuickBooks.

Steve: I didn't see that, but that's interesting.

Leo: Yeah. So maybe I'm mistaken, but I'm pretty sure I remember seeing that. The problem is all the reports we have are from local television news in San Francisco. Trust me, having worked with many of these people, not the most sophisticated. So we'll see.

Steve: Yeah. They're not hired for their technical savvy.

Leo: No.

Steve: We'll just say that.

Leo: And it does seem like it was ransomware; yes?

Steve: Yes, it was, yeah, $73,000 in bitcoin was the ask. And I don't know how it got resolved, whether they paid up, I mean, that's a lot of money. But who knows how expensive it would have been to, I mean, hopefully there's a backup plan and they would restore.

Leo: Well, that's what I would expect is that they, yeah, that they just restored from backups.

Steve: So it just created a glitch.

Leo: Right.

Steve: Speaking of glitches, Google's SHA-1 death march drumbeat continues. Now, it was right around this time, maybe a few weeks later, one year ago, everybody will remember that I was annoyed that I was being forced to move to SHA-256, but mostly because there were still systems which were not, that is, users' systems, many, that did not understand SHA-256 certs. And there were, you know, we looked at some statistics about - CloudFront did some analysis, and there was still a large percentage of machines, not in the U.S., but scattered around, for example, running XP. XP Service Pack 3 informed XP of how to do SHA-256. But any pre-SP3 XP didn't know how to do SHA-256.

So my concern was that, on one hand, I didn't want Chrome users to start telling me that GRC is not secure. At the same time, I didn't want to move to SHA-256 before I absolutely had to because I would be denying GRC's services to people who only had access, for whatever reason, to SHA-1-understanding clients. So my solution was, and DigiCert, my absolute favorite certificate authority, came to the rescue. Working with them, I was able to create an SHA-1 cert, actually both an SHA-1 and a 256, and so I had the SHA-256 on standby. And I didn't go right up to New Year's, but I went close enough because of course we always run across problems where calendars are off by a day or two. And so I didn't want that to happen.

So a few days before the end of the year I just gracefully switched from SHA-1 to SHA-256. And essentially what that allowed me to do, then, was to give all of the people who might be visiting GRC an extra year of time where Chrome wouldn't be complaining, and they would be able to connect. And I figured anyone still using SHA-1 who hadn't figured out a way around that, after many other sites would no longer work, GRC was no longer going to be standing out as a problem. So the next step of this is happening, I would say next month, except we're not quite in December. It's in January of 2017.

In a blog post that Andrew Whalley at Chrome Security posted, he said: "We've previously made several announcements about Google Chrome's deprecation plans for SHA-1 certificates. This post provides an update on the final removal of support." And just to give a little background and a little butt-covering: "The SHA-1 cryptographic hash algorithm first showed signs of weakness over 11 years ago." Eh. Oh, okay. I mean, there were some reduced-round attacks where, if you didn't actually do SHA-1, but you did an SHA.1, then, yeah, there were some problems. But SHA-1, you know, we're retiring it, not because it has a critical problem, but because it's prudent to do that. We would rather retire it before it has a problem than in a screaming emergency, if someone develops some silicon that's able to do those hashes.

So he says 11 years ago there were some signs of weakness, and "recent research points to the imminent possibility" - okay, I guess anything is imminently possible - "of attacks that could directly impact the integrity of the Web Public Key Infrastructure, or PKI. To protect users from such attacks, Chrome will stop trusting certificates that use the SHA-1 algorithm, and visiting a site using such a certificate will result in an interstitial warning."

So their release schedule, he says: "We are planning to remove support for SHA-1 certificates in Chrome 56, which will be released to the stable channel" - now, that's after developer and beta, then stable, that's what we all get - "around the end of January 2017. The removal will follow the Chrome release process, moving from Dev to Beta to Stable; there won't be a date-based change in behavior," which is to say it won't be - it'll be a version-based change. When Chrome 56 comes out, it will not know about SHA-1 certs.

And so he finishes, saying, "Website operators are urged to check for the use of SHA-1 certificates and immediately contact their certificate authority for an SHA-256-based replacement, if any are found." And then I said parenthetically here, "And, oh, by the way, the Vogon demolition and construction fleet will be arriving shortly thereafter. However, no one will have cause to complain about either of these events because proper notice of the end of SHA-1 and Earth's pending destruction have both been clearly posted in our offices at Proxima Centauri."

Leo: Now tell me how you like my poetry.

Steve: In other words, just wait till the end of January because Chrome is now the majority browser on the Internet. And we all know I reluctantly endorse what Google is doing. All of our lessons that we have learned teach us that it is only when we are forced to change, we do. There will be some blood in the streets because suddenly sites that are perfectly secure will be alarming their visitors, saying they are not secure; that the site uses obsolete cryptography or whatever jargon Chrome chooses to use. Anyway, [audio dropout] fun in two months, two months from now, toward the end of January, when Chrome 56 comes out. And with any luck, there will be a deferment to the Vogons' plans to put a spatial bypass right through where our comfortable little solar system is located.

Leo: To make room for an interspace bypass. Love it. Love it.

Steve: So this is an odd story, and I included it because a number of our listeners said, "What do you think about this?" You might want to bring this up in your browser, Leo. It's a site called Deseat.me, as in remove my seat, S-E-A-T. So D-E-S-E-A-T dot M-E. Fortune magazine mentioned it, which is what brought it to the attention of some of our listeners. It's an interesting idea.

And the idea is, their argument is, over the course of your relationship with Gmail, lord knows how many things you have signed up for over the years. And of course Gmail famously keeps the repository, this archive of all your email, unless you go to some lengths to expunge it. This thing, you log in with Gmail into Deseat.me, using OAuth. It then asks for access to your email, which is what makes me uncomfortable about this whole thing. But what they then say they do is read, automatically of course, process all of your email backlog and find all the things that you are still joined to or members of. And so, for example, I had four, Leo. And I'm seeing the number 283 on your screen.

Leo: You had four? My, you are very parsimonious.

Steve: Well, I'm not a Gmail user. I have it because, you know, Google. But otherwise, you know, I...

Leo: So this lets me sign out of these accounts?

Steve: Yes, with a single click. You're just able to go, oh, I don't need that, don't need that, don't need that, don't need that, don't need that, don't need that, don't need that.

Leo: That's intriguing.

Steve: So it is. So with the understanding that you are giving a third-party automated entity access to your historical archive.

Leo: Right.

Steve: That being the tradeoff.

Leo: Some of these I don't even know, I don't remember ever signing. I don't know...

Steve: Exactly.

Leo: I don't know what they are. Okay.

Steve: Yeah.

Leo: PriorityAuto.com? I don't know what these are.

Steve: Now, I don't know that spam wouldn't be in there.

Leo: I think it is because...

Steve: Because it was an American Express thing that they...

Leo: Yeah. I don't think I ever joined anything called FocusedPassion.com. So can you look at the - Turtle Tech Design? See, I think this is spam, too. I mean, it's clearly not discriminating.

Steve: So what they do is - their UI was a little unclear, but you start on the top. And so you just click on the disposition you want for the first one, and then that slides them up. And so you just sort of go through, saying, keep, dismiss, keep, deseat, deseat, deseat, keep, so forth, and just sort of walk through it. And then they provide the mechanism for essentially removing you from those lists or the memberships or whatever.

Leo: Yeah. I mean, some of these things I recognize. You know, what I just do - so it removes you from mailing lists, basically.

Steve: Yeah, I think that must be it.

Leo: So there's other things that do this, like Unroll.Me. You know what I do is I just filter on anything that has the word "unsubscribe" in it, and I put that in a folder called "Mailing Lists."

Steve: Nice.

Leo: And then if I want to look, I can look.

Steve: Yup, nice.

Leo: This clearly has spam in it, as well.

Steve: Okay.

Leo: And the problem I have with that is I don't want to unsubscribe from spam because they don't know I exist.

Steve: Right. Very good point.

Leo: So interesting idea.

Steve: Yeah. And that was my feeling. I didn't like it from a privacy standpoint. And they're also not quite forthcoming. They imply that this is secure because it's all being done on your computer.

Leo: Yeah.

Steve: So it's like, uh, I mean...

Leo: No, it's not, though.

Steve: I don't think...

Leo: It's going through my Google Mail.

Steve: It could have a big blob of JavaScript in the browser, and so the browser is doing that. Except when you and I both did the OAuth login, it did ask for permission for access to Gmail.

Leo: Yeah, yeah. In fact, now I want to take it away.

Steve: Yeah.

Leo: I've got to figure out how to get rid of it.

Steve: Yeah.

Leo: Oh, well.

Steve: So I got a tweet from our friend Simon Zerafa, who asked me to bring the Snoopers Charter Petition to the attention of our listeners. I know we have a large base of listeners in the U.K. So there is a Snoopers Charter Petition. It only needed 100,000 signatures, but when I made the show notes this morning there were 136,565.

Leo: It passed. I mean, it got enough signatures, yeah.

Steve: Yes. Although my feeling is the more, that is, Parliament will, I mean, it's overcome the threshold for bringing it to the attention of Parliament. But if it had a million signatures or two, then I would think that would carry more weight. So I wanted to make sure that anybody who is affected by the Snoopers Charter and felt strongly enough that the so-called Investigatory Powers Act, there is a petition. And I've got the link in the show notes. And this week I already put the show notes up on the Security Now! page at GRC so that people would be able to find it because it's petition.parliament.uk/petitions/173199. And I see that we're now at 141,244.

Leo: Which exceeds the 100,000 required. But, I mean, it's not binding, by any means.

Steve: No, no. But, you know, as an expression of annoyance. And, I mean, I don't know if that even matters.

Leo: Yeah, here's the response. "The Investigatory Powers Act dramatically increases transparency around the use of investigatory powers. It protects both privacy and security and underwent unprecedented scrutiny before becoming law." We couldn't pass it for years. Yeah.

Steve: Uh-huh.

Leo: Yeah. That is the story, that this is, in fact, even some independent people said, well, better this so you know what their capabilities are than not having prescribed investigatory powers, so everybody does whatever they want.

Steve: Well, and like the U.S. Patriot Act, which we found out was being way pushed beyond what its original legislation and legislators intended.

Leo: Right, right.

Steve: So, yeah, I mean, I can certainly see that having transparency is a good thing. And I guess you guys were talking about it on TWiT this Sunday because I...

Leo: Oh, yeah, we talked about it a lot, yeah.

Steve: ...remember hearing some dialogue that, you know, the concern, of course, being that this may be an indication of what is inevitable for democracies.

Leo: Worldwide. Not just us, but worldwide. And not just them, but worldwide, yeah.

Steve: Right. So Check Point Security, a well-known security company, named something that they found "ImageGate." You know, it's a gate, so Watergate and so forth gate. In this case, ImageGate. And the details are at this point scarce because they're withholding full disclosure responsibly until the social media sites that they have discovered with this problem. And they specifically note Facebook and LinkedIn, Facebook of course being huge, have fixed their problems. But it appears that, through some fault in something that Facebook and LinkedIn and apparently others are doing, it is possible for malicious executable files to be uploaded as images, which are then posted on Facebook pages and made available to unwitting users.

Now, as we know, the browser should just display the image. But the browser doesn't recognize this as an image. So when the user clicks on it, or clicks a link, up pops the "download this file" dialogue. So, I mean, so this isn't like a vast worm that's going to spread like crazy because it requires the user's involvement to click on, yes, save this, or open it, and then go find it and open it or look at it. But what's in there is an executable. And the point of all this is that there has been a rash of Locky ransomware spread through social media. This is how it's happening.

So what Check Point figured out was where was all this - how was all this Locky ransomware getting past Facebook and LinkedIn and other social media sites' protections. And again, there just isn't enough information. It'll be fun to cover this again when we know more. But for what it's worth, from what I could gather from what little they said, the fact that you have to ask your computer to execute it means - and also the fact that it's a way of getting it into Facebook. It must be some sort of a failure in the incoming file analysis which is missing the fact that an executable somehow disguised as an image is getting into the social media site and then being available for redistribution.

So they're described as Locky ransomware decoy image files which are ambushing

Facebook and LinkedIn accounts, as Ars Technica described it. So it'll be fun when we know more, when we can know more. Right now these sites were notified by Check Point, who have said, look, this is what's happening. And I'm sure they're in the process of locking that down, and we'll probably know more, maybe next week.

Oh, and Leo, this is a good one. The exploit is called Speake(a)r, as in Speaker, and then they stuck an "a" in parens in between the "e" and the "r" at the end. So instead of Speaker, it's Speake(a)r, which actually is kind of clever. And this has been making the rounds in the security community because the presumption has been, if a system doesn't have a microphone, or you've disabled, explicitly disabled the microphone, in much the same way people, for example, James Comey and Zuckerberg both have post-it notes over their webcams because they understand what can happen.

So it turns out that this group very cleverly realized that headphones, or speakers, are also microphones. And the catch is, of course, that you plug your speakers or your headphones into an output jack on your computer. Well, I'll just share from the abstract, just the top of their paper. They wrote: "It's possible to manipulate the headphones or earphones [or speakers] connected to a computer, silently turning them into a pair of eavesdropping microphones, with software alone. The same is also true for some types of loudspeakers.

"This paper," they write, "focuses on this threat in a cybersecurity context. We present SPEAKE(a)R, a software that can covertly turn the headphones connected to a PC into a microphone. We present technical background and explain why most of today's PCs and laptops are susceptible to this attack. We examine an attack scenario in which malware can use a computer as an eavesdropping device, even when a microphone is not present, is muted, taped, turned off, disconnected. We measure the signal quality and the effective distance and survey the defensive countermeasures."

So I love this story because this is one of those unintended consequences of the way our hardware has evolved. So first of all, as we know, many microphones and speakers [audio dropout] the same thing. Some are not. Back, I'm sure, Leo, being you're about my age - in fact, by the way, Happy Birthday.

Leo: Thank you.

Steve: Happy big six-oh today.

Leo: Thank you.

Steve: We unscrewed the mouthpiece...

Leo: Right.

Steve: Of the handset, and there was this wonderful - it was called a "carbon button microphone." And sometimes the quality, you know, somebody on the other line would say, "You're really scratchy-sounding. I really can't hear you very well." And you could knock that Bakelite handset on the counter or the desk and then say, "Okay, is this better?" And they go, "Oh, yeah, much better." Well, what you had just done was to shake up the carbon granules - I'm holding it to my head now, talking - to shake up the carbon granules in that microphone. And Edison coinvented that with somebody else over on the other side of the Atlantic. I can't remember his name. And Edison got a patent on it.

The idea was that you had carbon granules in an envelope, and the acoustic pressure from your voice squeezed the carbon granules together, thus transiently lowering the net resistance of the entire thing. So you had a conductive front and back. And then, if you talked at one of those, like talked into one of the, like the front layer, the vibration would squeeze the carbon, and you'd get a variable resistance, which is what the telephone company used back then in order to complete our calls.

Leo: But even regular headphones, speakers, it's a two-way thing. So the diaphragm that's in a microphone converts sound pressure from your voice into electrical impulses. But the headphones take the electrical impulses and convert them back to sound pressure. So if you plug any headphones, these headphones, into a microphone port, it's not a perfect microphone because it's not designed for that direction. But all headphones are microphones.

Steve: Yup. That's exactly right.

Leo: It's a two-way street.

Steve: Except electrostats probably are not. Essentially what you need is a moving...

Leo: Maybe no. Yeah.

Steve: You need a moving coil.

Leo: Right.

Steve: And so the idea is that today's, most of today's microphones are called "dynamic" microphones. And so there's a very strong permanent magnet and a very lightweight coil that is suspended sort of in air around that magnet, and then a front-facing diaphragm. Now when you talk, that coil moves back and forth, cutting through the lines of force of the magnet, which induces an electrical current in the coil to produce a current that is directly proportional to the displacement of the microphone's front surface. So we could think of that as a generator, that is, as you talk, as I'm talking into this thing right in front of my face, it's generating electricity - very, very weak, but that's what it's doing. And a generator you spin, like in a traditional motor generator, you spin the shaft of a generator, and out comes electricity. You put electricity in, and the shaft spins. That's a motor. And they are, as you said, the same thing.

So here's what happened. Once upon a time, on the motherboards, there would be a speaker amplifier chip, or a headphone amp on, like, connected to typically a D/A, a digital-to-analog converter. So there would be a D/A that would go to an amplifier that would produce the current to drive the headphones or speakers because a D/A typically, itself, it just produces a sort of a pilot current or voltage, but isn't strong enough directly to drive that. And you want a volume control and other features.

And then in the reverse direction there would be a mic amp where the very, very weak current from the microphone - and that's why, for example, microphones are so susceptible to hum, is that the induced 60-cycle hum in a microphone cord is large relative to the signal the microphone itself produces. So the microphone amplifier has to have a lot of gain in order to bring up the power from the microphone. In the process it brings up any hum that might be produced. But that was the architecture. You'd have a speaker amp outgoing, and a mic amp incoming.

Well, over time, with increasing levels of integration, because it was cheaper not to have multiple modules, that all got integrated, primarily by a company named Realtek. Realtek pretty much owns this market. They are way the majority chip in PCs and Macs that perform this function. And some guy at Realtek said, you know, sometimes our chip wants to do like a 7.1, sometimes a 5.1, sometimes stereo. So what they did was they generalized the chip. Instead of having inputs and outputs, they gave all the pins both functions. So it's 100% software defined. And the output is only an output because the BIOS programmed it to be an output, or the Realtek driver operating in the OS kernel said, oh, this system is set up to be this configuration. And so during boot time, or as the OS is coming up, it configures the hardware the way you want it.

What these guys realized was exactly that, and that it was no longer the case that any hardware statically defined inputs and outputs because the Realtek chips that are in most of our PCs and Macs are set up so that they are software-configurable. So if you've got headphones plugged in, or speakers plugged in, and something malicious gets into your system that wants to listen to you, and even if you have no microphone plugged in, if the machine doesn't have one, these guys demonstrated it's possible to turn the speaker into a microphone and listen to you just as if you had the microphone. Beautiful piece of work. Very cool.

Leo: Not even very surprising, frankly.

Steve: Yeah, but it's one of those things where, you know, I mean, it's like, okay, I'm safe because I don't have a microphone.

Leo: Right.

Steve: Uh, sorry.

Leo: You do have a microphone, actually.

Steve: You actually do. And it turns out you don't have to plug your headphones into the microphone jack. You just can redefine the speaker output as a mic input.

Leo: Right. And almost all headphone jacks now are single jacks with three rings for in and out. So your headphones are probably plugged into your microphone on most computers. You know, if you look at your computers, they have one jack now. Just like your phones. It's in and out.

Steve: Oh, interesting.

Leo: Yeah. There's three rings on your connector: one for a mic, one for left, and one for right.

Steve: Ah, okay.

Leo: All laptops, this one, I'm looking at the Creative Studio, it's got one jack. It doesn't have the - that's the old days where you had a Sound Blaster 16, and you had a microphone jack and a speaker jack.

Steve: So NTP, the Network Time Protocol...

Leo: I'm glad you're bringing this up, Steve. I saw this article, and I thought, we've got to spread the word.

Steve: So it's not a sexy protocol. Of all of the protocols around, it's like...

Leo: I think it's dead sexy. We all use it all the time.

Steve: Yes, yes. So it's sort of like ICMP. It's plumbing. It goes unseen. It's sort of taken for granted. And we've talked about it before. I'm impressed. I looked at it. And consider the challenge of obtaining super accurate time over an inherently nondeterministic network, that is, here's an NTP server wants to offer the time of day, as a server, to any number of clients that want to ask it. So it needs to get the time, like a reliable time, from a big daddy NTP server. Except that's going to be some number of router hops away, and there's an inherent time delay of at least milliseconds which can fluctuate and vary. Anyway, anyone who really has some time to burn, if you ever are curious, the NTP RFC, which goes into the technology, will not disappoint anybody who loves details because this thing, as simple as it seems on the surface, the problem they tackled was big.

Now, we haven't talked about NTP for a few years because I think it was like in what, end of 2014 or early 2015, it was for a while a popular DDoS reflection attack vector because it turns out that there were many NTP servers that you could generate a simple query to, which would result in a much larger response. And it's over UDP, that is, not TCP, the non-connection-based protocol, which means it can be spoofed. You can't really spoof a TCP connection because you require packets to have a roundtrip between the two endpoints in order to bring the connection up. UDP is used, for example, also by DNS, where it's just a single packet, which is low overhead, lightweight, much less expensive in terms of traffic on the 'Net.

So what has come to light is there has been a flaw for quite a while in all NTP servers. Every version up to 4.2.8p9, but not 4.3.94, are vulnerable. So before this came to light, because this was a concern, the NTP maintainers were notified responsibly, and they fixed it. The problem was just the receipt of a single deliberately malformed UDP packet could cause a null pointer reference, that is, could cause the NTP server to try to access the object at zero, which is always illegal and is always kept, it's always trapped and caught by the operating system because it's a mistake. And what the OS does is terminate the process. It's like, oh, you just misbehaved, and you're no longer a happy client of this OS, so go away.

And so what this means is that it would crash your NTP server. And so the concern was that NTP is everywhere. It's not, as I said, not very sexy. People don't talk about it. It just - it's a workhorse. But if this had been found and exploited, it could have brought down time, the end of time, for the Internet. It was fixed. But the takeaway here is that, because it's not sexy, people aren't dealing with it very often. I mean, it's like one of those things in the closet. So if you have a publicly exposed NTP server - and that's an important caveat. That is, for example, you might be running one on a Linux box or a Unix box at home. But it might be behind your router, providing time services to your own LAN.

Well, okay. So something malicious in one of your machines could bring down your LAN's NTP server, but that's probably not a big worry. The concern would be any publicly facing NTP services that had not been patched could be hacked and just crash. And in fact, now that this is news, and we know what the problem is, it seems very likely that someone will just get up to some mischief. They'll just scan the IPv4 space, spraying out malicious NTP UDP packets, crashing all of the still-not-patched NTP servers. So I just wanted to let our listeners know this has happened in case any of them haven't updated NTP to the latest. It's worth doing. Oh, and at the same time, there were 10 problems fixed. That was one of them. Nine less critical problems were fixed, as well.

Leo: And you can thank Harlan Stenn for that. And this is the - I thought you might be doing this, but this is the public service announcement I want to do. This is an article yesterday in InfoWorld. And credit to Fahmida Rashid for writing this. We all use NTP, but maybe you didn't know this project, which has been around for 30 years, has no corporate sponsor. Many open source projects do, and has one, count 'em, one maintainer, Harlan Stenn, who's basically doing this out of the goodness of his own heart. Now, this is an important open source project. It is severely underfunded.

Steve: Well, and we know about 2038.

Leo: Yeah.

Steve: Because the problem is...

Leo: The Unix time epic.

Steve: Yes. It goes to all ones, and then it goes to zero.

Leo: So the Linux Foundation gives it some money. It's part of the core infrastructure initiative. And when asked about vulnerabilities and why it's slow to fix them, including this one you just mentioned, Stenn says: "Reality bites - we remain severely under-resourced for the work that needs to be done. You can yell at us about it, and/or you can work to help us, and/or you can work to help others to help us." So if you are a coder, or you can help financially, the NTP Project needs your help. This is one guy. I mean, I think there are other contributors. But the maintainer is Harlan Stenn.

Steve: Well, and the good news is there isn't a huge demand for the spec to be evolved. I mean, it's not like TLS or something, which is rolling forward. It's like, this problem has been solved. But you're right, Leo, this is a perfect example of here there was an urgent need to get this fixed.

Leo: Right. And with one guy, you know, as he says, I'm doing the best - I'm working - it's that Venn diagram you mentioned earlier.

Steve: Yeah.

Leo: I'm working as fast as I can. So it's important.

Steve: And he said, "We're trying to do it." And I thought, well, he must have a mirror.

Leo: No, there are other contributors.

Steve: Oh, okay.

Leo: I shouldn't say he's the only guy doing any code. But he's the sole - he's the principle maintainer. And the way open source projects work, there's one guy who's like, this is my job. And then people come and go. And I think that's part of the problem is it's just one guy. So I'm glad you mentioned the bug. But let me also do the PSA, do what you can do help NTP because we all use it.

Steve: And why it still needs to be maintained is a perfect example here.

Leo: Yeah.

Steve: So it was probably foreseeable that the National Highway Transportation Safety Administration was going to give a little more attention to the problem of distracted driving. I just wanted to put this on everyone's radar. This is still in the early stages. But what is the acronym, the NHTSA is recommending a car mode which is somewhat similar to the airplane mode, but of course with a different intent. This one turns off all distracting apps which are not arguably needed for driving. So things like maps get to remain active, and presumably hands-free phone, but not Twitter.

So there is a 96-page voluntary guideline document which is intended - which specifies the intent of reducing driver distraction and, among other things, calls for cars to be more easily paired with mobile devices so that drivers can access their mobile devices through a presumably less distraction-prone in-vehicle interface. And so the guidelines go into some detail, suggesting, for example, that driver mode as envisioned would lock out things like typing out text messages, as well as displays of images or video not related to maps. It would block most text content, like web pages, social media, books and periodicals.

And, here it comes, the NHTSA says it's looking forward to developing technology that enables better "driver passenger distinction," presenting the possibility of a future in which phones automatically lock into some type of driver mode without needing the driver to initiate it. So anyway, it's not legislation yet. But, boy, Leo, I know you drive. I drive. I see there has been a, in the last few years, a dramatic increase in...

Leo: It's 10% year over year. I mean, it's terrible.

Steve: Yes, of accidents, yes. And I see, for example, lights turn green, and the cars don't move.

Leo: Yeah.

Steve: Because they took that opportunity to check in with Twitter or with...

Leo: They'll ticket you. My daughter got a big ticket for that, for not moving when the light turned because she was distracted.

Steve: Wow. Interesting.

Leo: I was glad she got it.

Steve: It is sort of a giveaway. It's a little hard to explain. Like, well, what were you doing when the light was green and cars were honking at you? And in fact, it's a little dangerous, too, because I've driven by some cars when the light was green, and they were just sitting there, you know, I mean, and they were in an adjacent lane.

Leo: Right. I'm busy here. I'm sending a text. Please.

Steve: And also, you know, I love to ride a bicycle. Thankfully, Southern California has given a lot of attention to bike riders. But I see cars now weaving, not staying in their lane, to a much greater degree now. And it's not just because of their automated driving systems. I'm sure it's because people are trying to do two things at once. And they can barely handle one thing at once.

Leo: Yeah, I think this was inevitable. And I'm not unhappy about it. I think the problem is what if you're a passenger?

Steve: Right. And thus their comment about, I mean, recognizing this notion of the driver/passenger distinction. Now, of course, you can imagine some driver then holding his phone out over in the passenger area and having to turn to look at it in order to get it to not lock down in driver mode. I mean, it is a problem. And look at the addiction so many people have to their phones. I see it, like, you know, every time I'm out eating. It's like a whole family is just sitting around, each one staring into their own phone.

Leo: It's really, it's kind of sad, isn't it, yeah.

Steve: It is, yeah. So the EFF generated a nice post. Unfortunately, they brought way too much of their own politics into it, which was unnecessary, you know, saying, well, saying: "The results of the U.S. presidential election have put the tech industry in a risky position. President-elect Trump has promised to deport millions of our friends and neighbors, track people based on their religious beliefs, and undermine users' digital security and privacy. He'll need Silicon Valley's cooperation to do it - and Silicon Valley can fight back."

Well, you know, fine. Unfortunately, my concern was that this heavily laden with politics posting would obscure what then followed, which were some really good recommendations, which are independent of who our President is. And that is for encouraging sort of generically the privacy offered by Internet-based services. And those recommendations are spot-on. For example, for sites to allow pseudonymous access: "Give users," they write, "the freedom to access services pseudonymously. Real-name policies and their ilk are especially harmful to vulnerable populations. Even better, don't restrict access to logged-in users."

And I won't go through all of this in detail. But stop behavioral analysis. Basically, don't try to infer things about your visitors that they don't voluntarily choose to share with you. Also delete logs. Logs are a privacy problem. So seriously look at what you have to keep, and only keep what you absolutely need to, and for no longer than it's actually useful. And then they remind us to encrypt data in transit, and also to enable end-to-end encryption.

So some nice policies that I think all of us would agree to. And let's hope that, I mean, unfortunately, Leo, I've heard you talk about this a lot on other podcasts, this also is a war that the end-user is losing. We are the product now to an increasing degree.

Leo: We're going to do a Triangulation in the near future with a guy named Tim Wu who's very famous, a Columbia professor of law. He wrote "The Master Switch."

Steve: Oh, great book.

Leo: Yeah. His new book is called "The Attention Merchants," and it's exactly that. It's the trade. And, you know, we do it, too. I mean, that's how we fund ourselves at

TWiT. We trade your attention to what we do with advertisers to fund it by running ads. But it's really gotten way out of hand. And so I'm looking forward to this Triangulation. I'm not sure what the date is, but I'll let you know.

Steve: Yeah, please do because I definitely want to watch that.

Leo: Yeah, he's very smart. And it's a great book, by the way. Highly recommended.

Steve: So here is just a deliciously detailed, which everyone knows I love, technical explanation of a mistake which, had it not been found and responsibly reported and corrected, could have allowed the malicious infection of more than one quarter, about 27% of the Internet's web pages, which are hosted on WordPress-based sites. So what happened? Every WordPress installation everywhere, by default, makes a request to servers at api.wordpress.org, about hourly, to just sort of ping api.wordpress.org to check for updates to plugins, themes, or the WordPress core. So that's often, but it's an inexpensive query to say, hey, you know, anything important.

And of course the argument is, if there were something horrible that was discovered, you'd want to be able to push that out to, in the core WordPress code, get that out to all of the WordPress sites distributed across the world. So I think that's a nice tradeoff, the fact that, about hourly, all sites, all WordPress running sites that are based on PHP just check in with the mothership. If updates are available, that api.wordpress.org server returns a link to a different download server which contains the update. And the WordPress site then uses the link it received from the api.wordpress.org update server to obtain an update for a code package and update itself.

The WordPress developers maintain their code at GitHub. To publish an update, once they're finished creating a set of improvements, new features or bug fixes or whatever, they commit those changes at GitHub. GitHub makes a query to the WordPress update servers to inform WordPress of the availability of new code on GitHub and to provide the URL, the GitHub URL from which the code should be fetched. Now, naturally, there's the potential vulnerability. That update notice from GitHub must be authenticated. Otherwise, anyone could create a malicious update package, inform the WordPress Update Server that it's ready for distribution, and thus potentially push a malicious update to every WordPress site - as we said, more than a quarter of the Internet - within an hour.

How does GitHub-to-WordPress authentication work? To start, both ends share a secret. So they have a shared secret. An update specification packet is created by GitHub. Then, for the purpose of "signing it," and I put that in quotes in my notes because it's not a formal crypto signature, but it's something these guys cooked up, and it seems cryptographically safe enough. That shared secret is temporarily appended to the update spec. And that concatenation is hashed using whatever hash algorithm is chosen. The shared secret is then removed from that spec - because, again, you don't want that to go out in the clear. And the spec and the hash are sent to the api.wordpress.org servers for verification and acceptance.

So once again, the spec says here's a new version. Here's the URL to obtain the updated code. And here's a hash of what we're providing which was made by temporarily adding a shared secret and hashing the whole thing. Then that shared secret was removed, and the spec, the update spec and its hash, which functions as a signature, are sent to

WordPress. So at the other end, at the receiving end, to authenticate, the api.wordpress.org server takes the specification, appends its own copy of the same shared secret, which of course when it hashes it, it's going to obtain the same hash that GitHub did. So hashes that; verifies that its hash matches the signature hash provided by GitHub. And that is the authentication mechanism based on a shared secret.

So it's simple, lightweight, and elegant. Since no third party knows the shared secret, no third party would be able to feasibly compute the proper hash to accompany a non-authentic malicious update. So what's the problem? The caller in this spec that WordPress developed for GitHub to use, the caller provides the update with the hash signature, as we said, and also specifies the hashing algorithm to be used <groan>. It's not by default SHA-1, migrating to SHA-256. No, it's actually a parameter in the query header that GitHub uses to send to WordPress. And that in and of itself is not a problem, except that this is all in PHP. And PHP supports many very old and low-security hashing algorithms.

And in the same category, even well-known, non-cryptographically secure algorithms like CRC32 checksum, and in this case something known as the Adler32 checksum. Now, we've discussed before, functions like CRC32 were never designed as a hash. They were designed and intended to detect mistakes in communication, not to prevent deliberate manipulation. But that's what a hash was meant to do. A hash is like a super checksum because not only would it find any change that was made by mistake, but you cannot, you know, it's cryptographically sound. You cannot deliberately make a change to the input of the hash in order to obtain a specific result. It just, you know, bits go in, and pseudorandom nonsense comes out the other end, of a fixed length.

Okay. So one of the things that this would do, you know, so the guys that were figuring out how to crack this, they said, okay, well, if we use CRC32, that dramatically weakens the system security. That is, if they said to WordPress, here's an update package and a hash, and by the way, please use CRC32, well, that's bad. So, I mean, that's not good because now you're saying use a function which essentially scrambles the input and produces a 32-bit output, rather than an SHA-1, for example, which is what GitHub and WordPress use, which is 160 bits, one six zero. So this has reduced the attack from 160-bit, which is really a lot, down to 32 bits. But still, that's 4.3 billion.

Now, remember that the attacker doesn't know the shared secret. So to exploit this, the attacker has to guess the shared secret, or essentially arrange to get a hash collision - where I use the word "hash" provisionally because CRC32 is not - but basically needs to cause the WordPress side to agree with the hash output, in this case 32 bits. But that's still 4.3 billion. And this has to be an active online attack, where continued update requests are being made to the server over a TCP connection, waiting, you know, trying. On average it would be half that many so, what, 2.15 billion. It still is going to take you a long time in order to get lucky.

Well, it turns out that there's something worse in this instance than CRC32, and that's that other function I mentioned, the Adler32 function. Now, okay, Mark Adler is well known, especially in the data compression industry. He's the coauthor of the zlib compression library and gzip. He contributed to Info-ZIP and also participated in the development of the compression used in the PNG, the Portable Network Graphics image file format. And as a little side note, he was also the Spirit Cruise Mission Manager for the Mars Exploration Rover. So he gets around.

He deliberately designed this Adler32 function, I think it was in 1995, so it's old. And he made it, you know, in all of these things there's a speed/quality tradeoff. Mark explicitly wanted something faster, rather than better. We already had CRC32. He wanted something faster. Well, a hash, one of the hallmarks of a hash is that all of its possible outputs are evenly distributed. That is, none of them occur more often than any others. And the way to say that algorithmically is that every single bit - like in the case of SHA-1, it's 160 bits. Every single one of those 160 bits has, for any given input, about a 50/50 chance of being a one or a zero. And there's no obvious inter-bit linkage. They're independent, and you just never know what they're going to be. That's what you want.

Well, that's not the property that Adler32 has. Adler32, because it was never intended to be a hash, it was just meant to be a very fast checksum, it turns out that its distribution of output values is massively non-flat. And these guys who found this attack realized that. And they were able, by asking the WordPress upgrade server to use the Adler32 function as the hash which was used to verify their submission, they were able to reduce the effective brute-force space down from 2^32, a full 32-bit space like CRC because that's the smallest of any of the functions that were available. But by using Adler32, they brought it down to between 100,000 and 400,000, which is entirely practical for an online attack. That is, like, what, 400,000. So an order of magnitude would be four million to - so it's like, what, four orders of magnitude faster than if it used a uniformly distributed function.

So they verified it. They built a proof-of-concept. They submitted it to WordPress. WordPress said, whoopsie, gave them a reward for their nice work, fixed the problem, pushed it out globally, everybody waited a while, and then these guys went public with their work. And a beautiful piece of work it was. So nice going, guys. And, by the way, that's WordFence.com are the people who did this. And they're a security firm whose whole focus is WordPress-related vulnerabilities and exposures and problems.

And so I'm glad they're keeping an eye on this stuff because, boy, if that had been abused by somebody, and we don't know that it hasn't been used in some subtle fashion to get some code into the WordPress system that could have caused problems. But at least we know now that this circuitous route, as a consequence of several factors that all kind of came together, created a vulnerability in a system that, you know, the spec looked good. But when you look at the whole thing from a profile of trying to attack it, there were some problems.

I did want to mention a little bit of miscellany. Supercapacitors are back in the news. A whole bunch of people sent this to me over the last week. Even Amber MacArthur picked up on it, Leo. I saw that she had grabbed it. And there's really nothing here, unfortunately. I mean, and there's not going to be until there is because, you know, we talked about supercapacitors years ago. I was very excited about a hopeful work happening down in the Southeast somewhere of the U.S. Well, in this case it's University of Central Florida, so same general area. I think the guys I was thinking about were in Texas for some reason. But remember we talked about it, the idea being that a supercapacitor stores energy as an electrostatic charge, rather than an electrochemical reaction. And unfortunately this was a really good photo and a very catchy headline because the headline was "A phone that charges in seconds." Except nobody has one.

Leo: Right. Somebody you'll be flying in cars.

Steve: And these people don't either. So it's just, you know, we need an advance in material science. We need - and it's going to be nano-something, and what was that, graphene, probably, in order to get high conductivity on a very, very thin dielectric insulator. And, I mean, I love the concept.

But the other thing that everyone skips over, and we've discussed it before, is that, because of the way capacitors store power - that is, they store it as a voltage. That's kind of the way to think of it. They store it as a voltage rather than a current. Power is the product of voltage times current. So you could have high power, either by having high voltage with low current or low voltage with high current. The high voltage with low current is the capacitor. The low voltage with high current is the traditional electrochemical battery that we're used to. And the problem is that our electronics of the day, it operates in a low voltage, high current mode. Like tubes, tube technology back in the day, when you and I were young, Leo, and we used to go down to the drugstore and check the tubes on our TVs because they weren't working.

Leo: [Crosstalk].

Steve: We'd, you know, plug them in and wait for them to warm up and glow.

Leo: I remember the tube testers, yes.

Steve: Yes. And you'd dial the knobs and things. I was fascinated by that. I thought, okay, someday I'm going to make one of these. Like, well...

Leo: I'm going to sell a tube tester myself.

Steve: Turns out I outlived the tubes.

Leo: Yeah.

Steve: So those were voltage-based, high voltage and low current. So they were sort of compatible with the capacitor as a storage mechanism. So it's going to be interesting to see, there's sort of an awkward wedding that we're going to need, if it ends up being that capacitors are the solution. I think it feels to me like it's the right answer because they are able to take current pretty much as fast as you can put it in there. But you need to give them high voltage rather than a lot of current over a long period of time. So we'll see how it develops.

And my last piece of crazy randomness is - oh, Leo, I think you're going to think this is really charming. It is free. It was a for-pay for about the first eight months, through August. It was released in January of 2016. It's called TraptionBakery, as in "contraption," but they left the "con" off. So T-R-A-P-T-I-O-N Bakery, B-A-K-E-R-Y. And it is not my normal type of puzzle. You download it for free on iOS, and it starts, and this framed picture fades onto the screen with, I don't know, some sneakers and a flowerpot in the foreground. If you scroll all the way down to the bottom - yeah. So there's the picture. I think if you scroll way down to the bottom of that website there's some animated GIFs - no one knows how to pronounce that. Okay, that's a different - there's a different link at the end of my show notes, I think.

But anyway, the point is it's an animated contraption, incredibly Rube Goldberg-like, with shoes that kick the cow that then eats the hay that decreases the weight of the basket that rotates the pulley that tips the flower pot that waters the plant and so forth. And all you do - yeah. And so that's a long page of him explaining the process of creating this masterpiece.

Leo: Oh, lord.

Steve: It really is amazing. But go way down to the bottom, and you will find where he's animated - there we go. And that gives you a sense of what this thing actually looks like.

Leo: It looks kind of like fun.

Steve: It is. No, Leo, I think this is a win. I just wish it were available on Android, too. So it's iPhone and iPad. And all you do is you zoom in, and as you zoom in, additional details reveal themselves. And the goal is to bake a loaf of bread. And so it actually is, you can - and he was trying, he explains on his page how he worked to be factually correct. He actually used baking principles of yeast rising and how much water to put in and everything. And then it evaluates the crust on the bread and so forth. But anyway, it's really - it's the kind of thing, you know, like if you're in a really boring meeting, you could just sort of poke around at. Yeah, so there you've got it on camera. And you just, like, zoom way in and...

Leo: What's the game? What do you do?

Steve: Well, that's part of the puzzle is...

Leo: I pulled that lever.

Steve: When you get close enough, additional things, and they show up in blue.

Leo: Yeah, yeah.

Steve: Like things that you're able to do.

Leo: Do you have to do them in a - I guess you have to do them in a particular sequence.

Steve: The whole thing, there's no owner's manual. There's no...

Leo: This is it.

Steve: It's just browse.

Leo: There's an elephant eating grain. Exit an elephant play area. Wow. This is - you know what I like? It's original.

Steve: It's charming.

Leo: And it's very original. It's like, wow. So it's really just a big drawing by Jon Prestidge.

Steve: It's a big drawing that is animated, that he put a ton of effort into. And some of the comments: One guy said, "While I'm not really much of a puzzle gamer, this one drew me in. Being able to figure out this contraption and make it work was one of the best puzzler experiences I've ever had. Everything, and I mean EVERYTHING," in caps, he says, "from the way the game is designed, to the artwork, all of the small details that become clearer as you zoom in, the thought behind every little thing and how it all interacts with everything else, is all absolutely perfect. I can't recommend this one enough to puzzle fans."

Leo: Yeah. Wow, this is really interesting. Huh. TraptionBakery. It's free, too, which is kind of cool.

Steve: Yes. It is free. There are several reviewers who said it was worth the money, and so I verified that - oh, and there are sounds.

Leo: It is worth the money - nothing.

Steve: No, no, I mean, back when it was for pay. And if you don't have sound turned up, sounds are a big clue, too.

Leo: Ah.

Steve: So it's making sounds which are important for your current view.

[Leo turns up sound]

Leo: Mm-hmm. Whoa. It changes colors. Maybe that's a bug, I don't know.

Steve: I don't think there are any bugs.

Leo: Oh, there are no bugs. Wow, this is a - headless people only. Oh. Oh, I made grain go in the thing. Aha. But now I can't blow the horn anymore. Hmm. This is cool. Very nice.

Steve: I think it's a win. I think it's - again, I wish everyone could play with it. But at least iOS, iPhone and - and actually, although you do a lot more scrolling with an iPhone, the fact that it's all about zooming means that a smaller screen can work, too, because you just have a smaller viewport into what's going on overall.

And, finally, Gary Nevills sent me - he tweeted an interesting picture. And he said: "Steve, do these patterns in SpinRite indicate anything specific about the condition of the disk?" And I don't think we've ever talked about this, Leo, but in the show notes I have a snapshot of SpinRite's graphic status display which has, like, a periodic appearance of mostly red U's, meaning despite everything SpinRite tried, it was unable to recover that data. But notice that, in a couple of the places where there would be a red U is a green R, saying SpinRite was able to - it initially could not read the data here, but it was then able, after working at it, to recover it perfectly.

What's going on - and I've been seeing these patterns for 30 years, I mean, since the beginning of SpinRite. Not often, but remember that spinning drives are still mechanical, and they inherently have a periodicity. They have tracks, and they have heads, and they have sectors around the track. So something has happened, probably one of the heads. This may be, I don't know how large the whole drive was because this was a piece of the drive. But, for example, if a head amplifier went bad, or if the drive - it could be a scratch on the surface. In the old days, you know, we called them a "head crash," literally a divot or a scratch on the surface.

Well, a scratch on the surface is going to create a problem, like a repeating problem in every sector that the scratch moves through. And you're going to encounter that on this track. And then you're going to go to switch to different heads. Then you're going to come back up to the same sector on the next cylinder. And that's going to be a problem again. So an infrequent, but periodically repeating, problem is evidence of a physical event, maybe the head, but more likely a head crash.

And what's really interesting is that sometimes we'll see them in one region. And the ends of them, like coming into the periodic, will be greens. Then they'll go all reds. Then they'll go back to greens. Meaning that the damage was less at the beginning and the end of the furrow plowed, so that SpinRite was able to recover over the damage if it wasn't too great. But in the middle, where it really became a divot, SpinRite just said, there's just, you know, there's a chunk of disk missing here.

So anyway, I thought that was really cool. So thank you for sharing that. And that's something I don't think we've ever talked about before is a spooky-looking pattern of repeating problems from a disk that clearly had some trouble. SpinRite did recover some little bits of it. But mostly it was really - and, by the way, if you ever see this, SpinRite is saying...

Leo: Get a new disk.

Steve: ...okay, get what you can off this. Don't give this, don't even give this one to your Drobo. Drobo will not be happy, either.

Leo: Well, what it looks like is like a single scratch, right, that's just...

Steve: Yes.

Leo: The reason it's repeating like that is just because of the nature of the geometry. But it's...

Steve: Exactly that.

Leo: Yeah, like a single scratch across the disk.

Steve: Yup. Isn't that cool?

Leo: That is cool. That's kind of neat, yeah.

Steve: Yeah.

Leo: We got Q. We got A. Questions and answers. Are you ready, Steve?

Steve: You betcha.

Leo: I am ready, as well. I have questions. You have answers. Let me push some buttons, get the Magic Q&A Eight Ball to give me your first question of the day. And it comes to us from Tim Chase, or @gumnos. @gumnos is his Twitter handle, G-U-M-N-O-S. And he asks: Regarding using my local WiFi interference to sniff PIN entry as a side-channel attack - by the way, this is the terse Twitter speaking here, so he's got this in 140 characters - my Android LG offers a "randomize the PIN order" function to mitigate. Is he talking about WPS?

Steve: No. So we talked about this, I think it was last week or the week before. I think it was last week. It was a side-channel attack which hackers had verified which used the MIMO antennas because there is metadata about the arrival time and relative intensity of the WiFi signal. And it turns out that somebody holding a smartphone, the motion of their hand was being conveyed by the WiFi signal. And with sufficiently robust recognition to weaken their entry of their PIN, the hotspot to which they were connected was able to figure out what their PIN was.

And so Tim noted that his Android LG offers a "randomize the PIN order." And so I thought that was a nice mitigation, which was actually meant to thwart a different attack, that is, for example, it's been noted that on phones or pads where you use a PIN, you can very often hold it in a certain way and see where the person's fingers have been. And of course a famous sort of apocryphal hack probably is after someone uses a keypad, you quickly take an IR photo of it, and you can see from the heat trail left by their finger the order in which they pushed the buttons.

And so what this does is, every time you're presented with a PIN, it randomizes what digits are where, which increases your level of effort because you have to search rather than it just being, oh, you know, 1234, which they're always in the same place, they're going to be wherever they happen to be. And don't use that PIN, by the way. But anyway, so just kind of cool because this was meant to thwart somebody watching you type it in, realizing, oh, I can see, like if it's 1234567890, even if you don't see the screen, you can kind of guess what the PIN was. And other things like leaving bits of peanut butter behind on the screen if you hadn't washed your hands. So, very cool. And as a side effect, this automatically mitigates a different attack that we were discussing last week. So very cool, Tim. Thanks for sharing that.

Leo: Clever idea. And credit to LG for offering it as an option.

Steve: Yeah, turn it on, if you've got it.

Leo: I think that's LG-specific. I've never seen that before.

Steve: And even better, don't use a PIN, use a longer passphrase.

Leo: Right. Right. Yeah, that's what I do. I use a seven-digit passphrase.

Steve: Yup.

Leo: Or a seven-digit password. Well, I still use a PIN, but it's seven digits. That's better than nothing. And you have only 10 tries; right? So here's another one from Domi, @420domi. Oh, boy. Which DNS provider other than Google Public DNS would you recommend if I want to capital "C" Change, capital "T" The, lowercase "d" default?

Steve: So we haven't talked about that much. But after the DYN attack, someone did mention to me that OpenDNS does what I was hypothesizing would be nice if someone did, which is to retain the previous IP address, even if the cached IP address expires. Don't refuse to give it out pending an update, but give what you've got. And someone said, hey, you know, Steve, OpenDNS does that. And of course OpenDNS does a lot of other really good things.

And so to answer Domi's query, and anyone else who's, like, wondering about an alternative, remember that DNS can play a much more active role in security than it normally does. DNS servers are typically generic, nothing added, just you ask them for an IP of a domain name, they go find it for you and bring it back. And then they remember it in case you ask them again, or anybody else does. But because of their role as an index to the Internet, that is, essentially you don't - the reason there was a big outage a few weeks ago is that the index went down. No one could convert domain names into IP addresses, and that's what our browsers need to make a connection.

So what OpenDNS has done, and they're not unique in this, but they've always been really good people, is they recognized they could add value to DNS. So, for example, they have a free offering called - first of all, you could just use their DNS servers, which are 208.67.222.222, and 208.67.220.220. So you just can use those. And they're stable. They use an anycast technology that we've discussed not too long ago, where the routers near you always send your queries to a nearby OpenDNS server. And they're like, they straddle the globe. They have servers all over the place. So they're serious about this.

But you can also, for free, use the so-called Family Shield program, which is preconfigured to block adult content. And so the idea is that, if anyone in your household attempts to look up, just visit a site, or maybe a browser by mistake tries unwittingly to obtain malware from a site that it shouldn't, it just says, sorry, don't know what the IP is for that. And so it sort of forms a simple, network-wide content filter for everything - your phones, your pads, your machines - because everything needs to use DNS. They also have a free Home version with customizable filtering, and they also offer ID theft protection. And then some inexpensive, $20 per year VIP packages.

Oh, and then the one last thing I wanted to mention is that DNS, as I referred to it when I was talking about NTP earlier, the Network Time Protocol, DNS just uses UDP. There's no encryption by default. I mean, there's not even any encryption in any spec for DNS. So one of the things that they offer is called DNSCrypt, which is a point-to-point, end-to-end encryption so that you run, on a Windows or a Mac, on those machines in your home, this DNSCrypt client. Your queries go to it locally. It encrypts them and sends them to OpenDNS. The reply comes back encrypted, and then it's decrypted for your use.

The beauty is that that creates privacy from your network, or at least from your machine and any machines that have that, over the Internet, so that no one is able to see what domains you're going to. Because, remember, even with HTTPS and strong authentication and privacy, your computer still had to ask its DNS server for the IP. Anybody looking at your traffic, you know, because DNS is not encrypted, they may not know what you do there, but they do know what you looked up, where you probably went. So OpenDNS. It's definitely worth checking out.

Leo: And you have, we should mention, you didn't plug it, but at GRC.com you have a DNS server test program.

Steve: Yes.

Leo: For free you can download if you just want to...

Steve: Well, I have the Benchmark, which is a Windows app that is also Wine compatible, so you can use it on Mac or Linux. And then I have a web-hosted service which is a Spoofability Test to make sure that the servers you're using don't have some common vulnerabilities. None of them should now. But it's interesting, there are some that are not generating very good random numbers, and that creates some weakness, as well. So, yeah, you're right, I've spent a lot of time looking at DNS.

Leo: And obviously speed isn't the only thing to consider, but it's an important criteria. It's knowing that, you know, the servers are fast, you can run the DNS Benchmark and then see. I've found that OpenDNS is very quick. They're really good about, you know, this is their business. That's all they do.

Steve: And up and stable. And I was a little sorry when Cisco purchased them. But they still seem to be doing a good job.

Leo: Haven't changed anything yet.

Steve: No.

Leo: We'll watch. Steven Throm, @dubious_rascal on the Twitter: I can't remember what hard drive imaging software you recommended recently. What was it?

Steve: Okay. So I get this all the time.

Leo: Me, too. I always - they say, "What does Steve recommend?" And I can never remember. So tell us.

Steve: Yeah. I'm going to put this - I'm going to move this - I forgot to do it - well, it didn't occur to me. I'm going to put it on the Link Farm page of things that everyone just can go to. It's Terabyte Unlimited is the company, and it's www dot terabyte, T-E-R-A-B-Y-T-E U-N-L-I-M-I-T-E-D dot com, TeraByteUnlimited.com. And it's called Image for Windows. They also have Image for Linux and Image for DOS, and bundles. It's inexpensive. It is my go-to imaging solution. I use it on my systems here. It's what I image the GRC servers with. It's able to run in the background. It's able to use the shadow volume copy feature in Windows in order to essentially obtain a static image of a system in use.

So, for example, Image for Windows, you can run it while the system is up and going. It will create an offline image of however many drives you wish it to, and/or partitions, and create a bootable thing. And then, if your life should end, and the drives should crash, and SpinRite can't help you because, I mean, it's like really, really, really dead, then Image for DOS, which is bundled with either of them, is bootable, and so it's able to read the image format and resuscitate, basically restore it either to the same or to a new drive. And it does things like partition sizing. It's got all the bells and whistles. It'll exclude the paging file and the hibernation file, I mean, it's a mature, beautiful piece of work.

Leo: Not free, but free to try.

Steve: Yes. Yeah, but it's not expensive. I think it's $30 something.

Leo: $38.94.

Steve: Yeah, it's a strange price, but still, yeah. And he's got [crosstalk] a bunch of goodies.

Leo: Part of the confusion is you went through a variety of different programs before you settled on this one.

Steve: I did. What was the other one?

Leo: You liked DriveSnapshot.de.

Steve: Drive Snapshot, yup, from the German guy.

Leo: And there's nothing wrong with that, now; right? I mean, that's still - yeah.

Steve: No, no, I just like Image for Windows.

Leo: Yeah. Nice, yeah. All right. One last short one, and then we're going to get to the long one. So this is Julius, @j_b_t: Prime solution IoT problem; TP-Link forces secure password input at first boot. Device doesn't connect/go live unless you do. Superb. Translate, please.

Steve: So this is Julius noting that he has a TP-Link router, and it is truly secure. It doesn't have a default password. It will not connect to the Internet until you have given it what it considers a secure password when it boots and you're configuring it, and you're not on the 'Net until you do that. And I just - I thought this was a nice indication, I mean, TP-Link is a well-known company. They produce a lot of great equipment. And I'm encouraged to see that kind of policy. It is a little more user-hostile because it's going to make you do something rather than just plug it in and go. But anybody who's setting up and configuring their own router, probably they just need a nudge to say, you know, do the right thing here. Instead of, like, "Oh, look, I plugged it in, Marge, and we're on. Let's go." So I'm not sure who Marge is, but still. Probably she's around here somewhere.

Leo: All right. Here we go. Get ready. This is going to be a marathon read. I have some work cut out for me here. This comes from Jonathan Nelson of Orem, Utah, about Malicious Internet Background Radiation, or MIBR: Hi, Steve. I had an experience a couple of days ago similar to the webcam being hacked you mentioned in Episode 587. I am a computer science student with a networking emphasis. My teacher issued us - ta ta ta tum - a MikroTik router in one of my classes so we can get hands-on experience working with routers and routing protocols.

The latest assignment required us to sign up for a 6to4 IPv6 tunneling service through Hurricane Electric since they offer a free service that can be subscribed to without having to call a customer service representative. I ran into problems when the tunnel setup process required an ICMP Reply to be returned by my router. I knew from using ShieldsUP! my router would not reply and couldn't get around this step.

I had the MikroTik router set up between my Windows computer and my router. One more, and I could have had a Three Dumb Routers setup. I looked in my router's firmware, an Asus RT-N16, for a setting that will let it reply to the ping, but couldn't find anything. I could have set up the MikroTik to be the exterior-facing router with my Asus as the interior, but that would have required a lot more configuration since the MikroTik had DHCP disabled as part of previous assignments. This also would

> have temporarily disconnected my wife - my wife, not my WiFi, my wife - from the Internet, an already not uncommon event due to my experimenting on my own, with which she has expressed frustration.
>
> I decided to set up the MikroTik as a DMZ connection. Immediately the ICMP packet went through and the reply back again. I was already logged into the terminal of the MikroTik router and had only typed the first three of four commands required for the assignment when I got a message on the console saying "system, error, critical login failure for user root from 79.66.93.197 via telnet." At first I thought I must have entered a command wrong from my homework. Turned out someone was pinging the router and trying to log in via telnet.
>
> I wasn't concerned, as I had set a password for the telnet client, but was surprised at how quickly the attacks had started. The IP address the attacks were coming from changed every so often. The username rotated between "admin," "root," and, curiously, "666666," among a few others. I finished my homework quickly and disabled the DMZ feature of my Asus router. This is the first time I have seen an active attack as it happened. Probably it won't be the last since I intend to get a job in network security when I graduate.
>
> Thanks for the podcast. I have listened for many years. Often as my teachers are explaining basic crypto concepts, such as preshared versus public key, I think to myself, "Duh, we all learned this before," and then realize, oh, I learned it from Security Now!. Thank you. P.S.: Feel free to rake me over the coals if I was stupid. I can take it.

Steve: So no stupidity. Sort of some cleverness. So the short version is he, for a very impressive-sounding class, needed to use a service in the cloud, a cloud-based, you know, Internet-based service that required a confirmation of the router's existence. It had to respond to unsolicited traffic, to an incoming ping. So the ASUS router he had on his border wouldn't do that, couldn't be configured to do that. So he used the DMZ feature of the ASUS to send unsolicited incoming traffic to a specific IP on his LAN, which is where the MikroTik router was configured. So that meant that any unsolicited traffic would go to the router.

The cool thing is that he happened to have - he was logged into the router with the console open when, as he put it, malicious Internet background radiation, because of course we know that I coined the term Internet Background Radiation, IBR, to sort of reflect the fact that there's just all this junk out there, packets flying around everywhere from, like, copies of Nimda and Code Red worm that are never going to die because they're in some closet somewhere, and they're still out scanning, thinking that it's, you know, partying like it's 1999. So basically, by using the DMZ on his border router, he exposed a router that had an open telnet port to the Internet. And within seconds something was trying to log on. And not just one something, but as he said, the IP address kept changing. That can't be a spoofed IP because telnet is a TCP protocol.

So those were real IP addresses of things, of something or other, you know, who knows, other people's DVRs or webcams or baby cams, scanning the Internet, looking for company. And they thought they might be able to nestle into his MikroTik router. He happened to have a console session open and got an error message, "You got the password wrong." He says, "Wait a minute, I didn't try to log in." No, some bot somewhere on the Internet did. That's today's reality, which is just amazing. So a very cool story.

Leo: Presumably there are, right now, all the time running, scanners scanning ports. One of the ports they look for is open port 23. Ah, that's the telnet port.

Steve: Twenty-three, baby.

Leo: Maybe some moron has put his telnet server online without changing the default password. Let's see. But I'm sure that goes on constantly.

Steve: Yeah. And that's of course why these things were trying admin and root, and there must be a bunch of devices that are 666666 as their default password.

Leo: Right, yeah. What do you think? One more? Two more? How do you want - what do you want to do?

Steve: Let's wrap it up, and we'll continue next week.

Leo: Dealio.

Steve: Perfect.

Leo: Steve Gibson and Security Now! every Tuesday, 1:30 p.m. Pacific, 4:30 Eastern, 21:30 UTC, if you want to hear your kind of college-level course, maybe graduate level course in the computer, in Internet, in security. You can get it at his site, GRC.com. Actually, while you're there, pick up a copy of SpinRite, world's finest hard drive maintenance and recovery utility. He's got lots of freebies there, including things like that DNS Benchmark. He just writes stuff. He's very prolific, and leaves it there, and then you never know. ShieldsUP! would be a good one to test your ICMP ports.

Steve: I just heard that the Guardian referenced GRC. Somehow they found my Cookie Forensics page.

Leo: Perfect, yeah.

Steve: Which, you know, to check the way your browser handles cookies. And people sent me tweets saying, hey, did you know that you're in the Guardian? It's like, yeah, cool.

Leo: That's the thing. All this stuff is there, and it just kind of stays there forever. So GRC.com. And browse around. You'll be amazed at the depth and breadth of stuff that's there. You'll find audio for this podcast - written transcriptions, too - for every episode, all, what is it, 588 shows. You'll also find the show notes there. That's the only place we put the show notes. If you want video, we also have that at TWiT.tv/sn on our website, TWiT.tv/sn. And it's on YouTube. It's on everywhere - Stitcher, Google Play Music. Anywhere you can get a podcast, you'll find Security Now!. And most of those places will let you subscribe so you get it automatically every week, which is a great idea. Start building your collection. Collect all 588. And then continue to collect.

We will be back next week. Actually, you're going to be up here Thursday.

Steve: Yes, I am, yup.

Leo: So 2:00 p.m. Pacific, 5:00 p.m. Eastern, 22:00 UTC. You'll be able to watch, because we were debating whether we should do it in secret and then surprise you on...

Steve: Ah. So it is going to be aired.

Leo: Yeah. It's for our Christmas Day TWiT. Nobody wants to do a TWiT on Christmas Day, so we thought this would be fun. It's something new. And it started, and you were there, one of our New Year's Eve broadcasts, you were there, Randal Schwartz, Paul Thurrott, I think. I can't remember. But we had three or four...

Steve: We had a really great roundtable.

Leo: Such a good conversation. And Lisa and I both remembered that and thought, why don't we do that every year, bring in different hosts? Next year it'll be a different group. But cross-pollinate a little bit. So we invited a bunch of hosts, and you were the first three to respond: Denise Howell, Steve Gibson, and Rene Ritchie. All will be in-studio with me.

Steve: Nice, nice.

Leo: And we'll be - what we're going to do is go through the big stories of 2016.

Steve: Perfect.

Leo: Yeah. And we won't want more than 10 or so, but it'll be a fun one that you can watch, if you want, on Thursday, and listen to and watch on Christmas evening, as well, because that'll be a lot of fun. So that's our Christmas episode this Thursday, with Steve coming up here. I'll look forward to seeing you. We'll save a bottle of Cab for you.

Steve: Perfect.

Leo: All right. Thanks, Steve. We'll see you next week on Security Now!.

Steve: Right-o.

Leo: Bye-bye.

Steve: Bye.