
# Messenger, CryptoDrop, & Riffle

**Description:** Leo and I catch up with a fun and interesting week of security happenings, including a bit of daylight on the password sharing question; the trouble with self-reporting security breaches; trouble in TOR-land; what future AI assistants mean for our privacy; a terrific-looking new piece of security monitoring freeware; a startlingly worrisome 20-year-old fundamental Windows architectural design flaw; a problem with Juniper routers' OS certificate validation; some errata; a bunch of miscellany; and the promised follow-up dissection of Facebook Messenger's extra features, the anti-ransomware CryptoDrop, and MIT's "Riffle" anonymity-enforcing networking solution.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-569.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-569-lq.mp3

---

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. He's champing at the bit, ready to go with a great show. Lots of security news. We'll talk a little bit about the 50th Anniversary of Star Trek and the movie to come. And then we'll analyze some security threats and some security promises. One's good, one's bad, and one's right in between. You'll find out. Details to come, next on Security Now!.

**Leo Laporte:** This is Security Now! with Steve Gibson, Episode 569, recorded Tuesday, July 19th, 2016: Messenger, CryptoDrop, and Riffle.

It's time for Security Now!, the show where we cover the latest news from the security front. And there he is, your Commander in Chief, on the front lines every day, Steve Gibson of GRC.com. I salute you.

**Steve Gibson:** Yo, Leo.

**Leo:** Hey, Steve.

**Steve:** Good to be with you again, as usual. I turned my salute into the Vulcan handgrip. And I was thinking, yes, and I'll be front row, well, not front row, but exactly the right seats on Friday.

**Leo:** Oh, you're going to the show.

**Steve:** I am.

**Leo:** Are you excited? "Star Trek Beyond."

**Steve:** I love the movies.

**Leo:** I do, too.

**Steve:** That is, I mean, the Star Trek movies. I really think...

**Leo:** Is it true that one is either a Star Wars fan or a Star Trek fan? You could be both. I'm more of, I'll be honest, I'm a Star Trek guy. I'm not a Star Wars guy.

**Steve:** Well, this is the 50th anniversary.

**Leo:** Wow.

**Steve:** The first series began in 1966, when I was 11, and you were either nine or 10. And it had an outsized influence on me. The quality of the original Gene Roddenberry stories and, literally, the characters that have lived for 50 years, ever since then. And actually, down in our miscellany, I have a little note about that because Netflix has an announcement that they just made that I will share.

But we've got a crazy bunch of news this week. A bit of daylight on the password-sharing question that we looked at last week. There's another result from a different three judges on the Ninth Circuit Court. The trouble with self-reporting security breaches. Some trouble in Tor Land. What future AI assistants mean for our privacy. A terrific-looking new piece of security monitoring freeware that you and I asked for, and one of our listeners delivered. A startlingly worrisome, 20-year-old, fundamental Windows architectural design flaw. A problem with a Juniper router's OS certificate validation. A bunch of errata, miscellany. And then the title of this podcast is "Messenger, CryptoDrop, and Riffle," which sounds like a law firm.

**Leo:** I like it.

**Steve:** Or at least the Riffle part, I guess. Anyway, I promised to follow up on the details of those when they exploded underneath me last week, and I thought, okay, we already have two hours. There's no way I could spend any more time. So Messenger has neat new features. CryptoDrop went from, eh, I don't know to, okay, where do I get one? And Riffle, unfortunately, which the press is saying, oh, an alternative to Tor, it's like, uh, no, unfortunately, the Internet really fights anonymity, and no one has figured out how to fix

that, even these guys. So lots of stuff to talk about.

**Leo:** Can't wait. And I don't have to wait long because the show is on the air. And I do, at some point I'll - I ordered my FreeBSD box. Now, not my server, my desktop FreeBSD box. And before the show I was saying I'm now ready for you to do a show on ZFS, the file system that is not only part of BSD, but is rapidly becoming part of the Linux world. ZFS for Linux is now kind of rapidly coming on strong.

**Steve:** Yeah, if you give it a lot of storage and a lot of RAM. It's RAM-hungry.

**Leo:** So I've got 32GB.

**Steve:** Good.

**Leo:** And I think that's enough.

**Steve:** Good, yes.

**Leo:** I looked at what the - because what I'm going to do is I have five drives. I have an M.2 drive. I don't think I'm going to put that - I think I'm going to put that in a separate pool. I have to talk to some BSD experts, maybe. I'll install, I'm thinking, the operating system, the ports, all of that stuff on pool one, which will be the NVMe drive, the fastest drive. And then I have four terabyte SSDs. And I'll make that the second pool. And I'll make that the home directory and the data stuff.

**Steve:** Nice.

**Leo:** And I'll use RAID-Z, their RAID thing for data redundancy, which will give me three gigs out of the four, or three terabytes out of four.

**Steve:** Right. And it handles drive degradation very nicely.

**Leo:** Apparently.

**Steve:** You can pull the drive out and slide another one in, and it just rebuilds it.

**Leo:** Yeah, yeah. And I don't need to have that kind of redundancy on the operating system and apps. I need it...

**Steve:** Correct. In fact, the architecture is very much what I described for the big box I built. I'm booting from an M.2, the Samsung 950. And then I have a pair of mirrored

multi-terabyte drives. And so what I do is I make a nightly image from the SSD over to the big RAID, which is just a mirror, so it's just, you know, simple redundancy because the problem with non-hardware-assisted RAID is it is a little slower. So anyway, I'm just using mirroring, so it writes the same thing to both drives. And that way I'm always able to back up to a previous image. And I keep a few images rotating, and then of course offsite backup. And we'll have a new sponsor next week that has something to say about this.

**Leo:** Oh, yeah. You're right.

**Steve:** Yeah. And I've got - I did get the whitepaper. And I need to dig into it because they've got, like, multiple layers of meta-ness around their RAID so that you get this sort of "it just works" sort of functionality. So I will figure it out so we can talk about it intelligently when they become a sponsor.

**Leo:** Yay. The other thing I'm doing as kind of an experiment is creating my own cloud on another server, a Linux server, using something new called Sandstorm.

**Steve:** Heard you talking about that on MacBreak, yeah.

**Leo:** I'd be very curious, your thoughts on that. It's intended to be highly encapsulated and secure, the idea being that, well, maybe FreeBSD doesn't have all the easy GUI apps. But if you can run a browser, I can do all the browser apps on my cloud, and then have access to it from all my other systems. I don't know, I think this will be - you know what? As I said, and you probably heard me say this, this is my game. And you're the same way. This is fun. This is something I look forward to. I was thinking about it last night and rubbing my hands. We're just weird that way, aren't we.

**Steve:** So we talked last week about the press pretty much going crazy over that decision, over their interpretation of a decision to uphold a prosecution of this guy that left a headhunting firm and then got the password from someone who was still there and clearly broke the law in breaking into that company's confidential data using this password. And so the way the majority opinion was written by those three judges in the Ninth Circuit, you could, I mean, in the best interpretation, they created some gray area where it really did seem like, if you shared your Netflix password with someone, you could be in violation of the Computer Fraud and Abuse Act, which to everyone felt like, okay, wait a minute, there's something wrong here between giving my girlfriend my Netflix password and going to prison.

So the EFF picked up a second decision made by a different set of three judges on the same Ninth Circuit Court of Appeals, which, as EFF said, backs away from dangerous password-sharing decision, but still leaves even more confusion about what the CFAA, the Computer Fraud and Abuse Act, actually means in the law.

What the EFF wrote was: "Three judges of the Ninth Circuit Court of Appeals have taken a step back from criminalizing password sharing, limiting the dangerous rationale of a decision issued by a panel of three different judges of the same court the previous week. That's good, but the new decision leaves so many unanswered questions that it's clear

we need [what they called] 'en banc' review," which means all of the judges.

**Leo:** Oh, like in the bench, by the bench.

**Steve:** Right, all 11 judges, not just three. So that would allow the court to issue "clear and limited interpretation" of this notoriously vague federal hacking statute which is at the heart of both cases, the Computer Fraud and Abuse Act.

And then the EFF said: "To recap, the court's language in last week's case" - that is, the week previous - "U.S. v. Nosal, was so broad that it seemed to make it a federal crime to use someone else's password, even with their knowledge and permission. In the new decision, in a case called Facebook v. Power Ventures, a separate Ninth Circuit panel acknowledged that a computer user can provide another person with valid authorization to use their username and password." Which that comports with how we would think this should be interpreted. Like if someone who has authorization gives theirs to someone else for their use, then okay. But again, still there's some gray zone.

So they said: "That's the good news. But the decision leaves unanswered so many other questions about how the law can be interpreted, and its rationale is so confusing, that it's an invitation for more dangerous litigation and prosecutions under the CFAA. The CFAA makes it illegal to engage in 'unauthorized access' to any computer connected to the Internet. But the statute doesn't say what 'authorized access' means or make clear where authorization must come from." And we've talked about this many times. This is the real problem when laws are left vague.

And I'm sort of now fascinated by the political process that runs this country, and so I've watched laws that can only be passed when some of the teeth is taken out of them. And so compromise in Congress ends up taking what was originally - you can imagine the original drafters may have had very clear definitions. But they were unable to get it to pass if it was worded that way. And so it went through committee and went back and forth and around and around. And it got, like, watered down so that it was finally weak enough that they could get a majority. But then it's also, unfortunately, then, imposed upon the legal system to figure out what this language means when all of the specification was deliberately taken out of it. So that's our system.

Anyway, speaking of our system, there was another little bit of news that just sort of put me in mind of a fundamental problem we have downstream of these high-profile hacking attacks. And that was there was a little blurb that the FDIC, the Federal Deposit Insurance Corporation, was hacked by China, and the CIO of the FDIC covered it up. This was in Ars Technica. Sean Gallagher reported last Wednesday. He wrote: "A report published by the House Committee on Science, Space, and Technology today found that hackers purported to be from China had compromised computers at the Federal Deposit Insurance Corporation repeatedly between 2010 and 2013. Backdoor malware was installed on 12 workstations and 10 servers by attackers - including the workstations of the chairman, chief of staff, and general counsel of the FDIC."

Now, the problem is, if that's you, you're like, oh, do we have to tell on ourselves? So Sean continues: "But the incidents were never reported to the U.S. Computer Emergency Response Team" - that exists to received such reports - "or any other authorities, and were only brought to light after an Inspector General investigation into another [different] serious data breach at the FDIC in October of 2015," so just last October. And that one was - and I cut it out because it was peripheral, but it was employees.

Half the employees, apparently, have access to USB thumb drives, and they're wandering around with hundreds of thousands of people's credentials and private information on their thumb drives. Because it's like, oh, here, stick this in, transfer this database, and I'm going to go walk across and plug it in over here. But it stays on the thumb drive unless you explicitly remove it, and so, whoops, a little bit of an information leakage problem. So 160,000-some-odd people have free credit reports for a year. Like, okay, well, thank you.

**Leo:** Well, there's a bright side then, yeah.

**Steve:** Anyway - huh?

**Leo:** There's a bright side.

**Steve:** Yes. So "The FDIC failed at the time of this advanced persistent threat to report the incidents." Failed. Whoops. Just forgot. It's on my to-do list. I promised, but I must have just skipped that checkmark. Then, the "Then-Inspector General at the FDIC, John Rymer, lambasted FDIC officials for failing to follow their own policies on breach reporting. Further investigation into those breaches led the committee to conclude that former FDIC CIO Russ Pittman misled auditors about the extent of those breaches, and told employees not to talk about the breaches by a foreign government" - remember this is China crawling around in the Inspector General's workstation, the chief of staff and the chairman - "so as not to ruin FDIC Chairman Martin Gruenberg's chances of confirmation."

And so anyway, stepping back from this, I just thought, you know, I mean, this is a problem because we're inherently asking the higher ups to obey their own rules, and there are guidelines, and they're written down. But, I mean, the only possible solution is to arrange, for example, not for employees who report to the people who would get in trouble to be responsible for finding this and then being told not to, but to somehow have completely disassociated auditors. And I guess auditors were involved, but the CIO downplayed it.

Well, we need a mechanism whereby auditors aren't taking clues from the C-level executives, but going in themselves. And again, clearly this is a function of - we saw what happened with Sony, the advanced persistent threats there, and the RSA - both companies hugely damaged, suffered major reputation damage. And of course in the government heads will roll. So we need some sort of mechanism where the lines of reporting are different from the lines of auditing. Otherwise you're just going to have, I mean, what employee is going to go against his C-level boss's wish and risk being terminated?

Now, we've got two things to talk about with Tor. The first is just some news that is a little vague, and its vagueness is a little disturbing, sort of for reasons of warrant canary concern, and involving the Tor project. We know that early last month there was a - one of the major previous Tor personalities, Jacob Appelbaum, resigned from Tor amid some scandal, which he denied, but he left anyway. Well, now we have Lucky Green, whom we've referred to from time to time through the years when we're talking about Tor. He was one of the very early contributors, before it had a name even, and has been running a number of important sort of key core servers, one called a "bridge authority." So anyway, yesterday he posted this note. He said: "Dear friends."

**Leo:** Sounds like a [crosstalk] problem. Sorry, I'm interrupting you.

**Steve:** "Given recent events, it is no longer appropriate for me to materially contribute to the Tor Project either financially, as I have so generously throughout the years, nor by providing computing resources. This decision does not come lightly. I probably ran one of the first five nodes in the system, and my involvement with Tor predates it being called 'Tor' by many years. Nonetheless, I feel I have no reasonable choice left within the bounds of ethics but to announce the discontinuation of all Tor-related services hosted on every system under my control." And there again it's like, okay. "I have no reasonable choice left within the bounds of ethics but to announce the discontinuation of all Tor-related services hosted on every system under my control." So we don't know what he knows. We don't know what has happened. There's no necessary linkage between the Appelbaum resignation a month and a half previous, but - so who knows.

And then his note continues: "Most notably, this includes the Tor node 'Tonga,' the 'Bridge Authority,' which I recognize is rather pivotal to the network." And the reason that is, by the way, is that bridge authority nodes have their IPs static and embedded in Tor applications. They're sort of like the roots. And so he's talking about terminating one of the first five nodes. In this case it's a bridge authority. So over time Tor apps need to change, unless somebody else can run a bridge authority server on the same IP, which doesn't seem to be happening.

So, he said: "Tonga will be permanently shut down and all associated cryptographic keys destroyed on August 31st. This should give the Tor developers ample time to stand up a substitute. I will terminate the cron job we set up so many years ago at that time that copies over the descriptors. In addition to Tonga, I will shut down a number of fast Tor relays. But the directory authorities should detect that shutdown quickly, and no separate notice is needed here. I wish the Tor Project nothing but the best moving forward through those difficult times." And then he just signs it "Lucky."

And in his posting there were some questions following up. One asked, like, what happened? And it's received no answer. So it's kind of hard to know. This is sort of reminiscent of TrueCrypt, where, again, we're left to speculate and try to come up with a theory that makes sense. But we have no absolute knowledge.

I read a story that I just sort of wanted to comment about a little bit. And this was in Tech Crunch. The title was "Why the top five tech companies are dead set on AI." And I was a little less interested in that, I mean, well, we know because that's where growth is. In the story they talk about how the pace of smartphone and desktop hardware innovation has slowed. My position is it's inevitable. That's the way the world works. And I thought of cars. Once we figured out that most cars should have four wheels, we're pretty much done with that question. Yes, you could have three, and you'd call it a tricycle, or two, or more. But most cars have four. Four seems to be the right number. We're kind of done with that.

And similarly, how long have we had a pretty much unchanging graphical user interface mouse and cursor model? Since Xerox PARC showed it to Steve Jobs. And it hasn't changed. You move the puck, and the little pointer moves around because it's done. And I always think of word processing and Microsoft Word, where they got it, like, a long time ago. It was just fine. Word 2003, Office 2003, works great. But of course they have a new version every year or two despite that.

Well, of course, on the AI front, we've got Apple with Siri, Google with their Assistant,

Amazon with Alexa, Microsoft with Cortana, and Facebook with their Chatbots, all sort of moving into this AI realm. And as I was thinking about this, what occurred to me is that there is a privacy impact because, with AI, context is king. And those of us who've been around sort of in the early days will remember the early speech recognition software which needed to be trained by having the user read a word list. That was part - remember Dragon Speak was one of the very early packages. Even back on the Apple II, I mean, it was a couple boards full of extra processing in order to get enough horsepower to do that. And I remember that Jerry Pournelle was a big fan of speech recognition and liked dictating stuff sort of in that era, in the CPM S-100 bus days.

But the way you trained it to be reliable for you was to read a word list and to interact with it and correct it. So it had to learn the way you spoke. Similarly with handwriting recognition. That's gotten a lot better. It is smart enough now to be more generic. But the early handwriting recognition was very similar. It needed to adapt to and learn the unique style of its user. And I think, because those are mechanical things, this thing wants to learn to understand me, that seems benign. Or this thing wants to learn how to read my handwriting. Oh, that's good, that's a convenience. But we don't think of those in terms of privacy.

But when we stop to think about it, it was an early form of biometric parameter acquisition. So now we move into the personal assistant space. And so then what's the context against which our needs will be understood? The context is everything available about us, our entire lives, as much as it has access to, because the more it knows about us, whatever the service is, the better job, the argument is - and it's a true argument, it holds up - it has to understand us in order to be able to, not only predict, but also to deobfuscate, in order to understand what we're saying or what we might mean based on our current location, for example, and based on the places we have been recently, based on what we've been doing on the Internet.

I mean, the more information, the more watching of us that can be done, the better job something, an assistant of some sort, can do of meeting our needs. So our lives and habits and interests are going to be recorded and profiled and modeled in order for this next generation of personal digital assistants to be able to perform their jobs. And it's happening now. And much as with speech recognition and handwriting recognition, maybe we don't notice or care because the benefit offered is clearly, in terms of cost benefit, outweighs whatever concern an individual may have over that kind of sort of explicit "I know what you're thinking" capability may be.

But I just sort of - this all played through my mind when I was reading about this story about AI and its future. Again, not to run around screaming with my hair on fire. But for our audience it's just worth considering that, the way these things work, in order to be as good as they're going to get, they just need to be sucking up, vacuuming every tidbit they can get about us. And with the justification that what we get in return is a more effective servant, a more effective assistant. Be interesting to see how this evolves. And I imagine there'll be some people who would rather opt out, stay off of that grid.

Okay, now, Leo, on Security Now! #551...

**Leo:** Oh, I remember that. That was a great episode.

**Steve:** And in fact I have our dialogue here. We recorded it when I was still 60 years old, on March 17th.

**Leo:** Ah, the good old days.

**Steve:** Leo says: "It's essentially a man in the middle sitting on my own machine, much like the certs from antivirus companies. Do you know" - it's Leo asking me - "of any tools that monitor certificate stores and report when changes are made?" I reply: "So, if I weren't so far behind, I would immediately whip out a utility. I'm not going to." And I don't know if that was before or after Never10, but it was right around that time. So maybe I was already whipping something out. Leo says: "That's a great idea, yeah." And I reply: "It is a great idea." Then Leo said: "Someone should write that."

**Leo:** Somebody has.

**Steve:** And I said: "And that's why I put this in this Q&A. Remember that Mark Russinovich just recently updated the Sysinternals tool so that, with a command line, it'll do that. Somebody could write a little Python frontend to the Sysinternals tool that is invoked by the scheduler" - this is me talking, not what we actually got - "or maybe just runs in the background and checks every day. If anyone does, make sure you send me a note, and I will make you famous. I will tell everybody" - everybody, you're being told - "about it because that ought to exist."

Well, it's called CertWatch, by a guy named Beau Blaser, B-L-A-S-E-R. And it is a beautiful-looking little piece of freeware. Now, I gave it the bit.ly link of the week, but this time without a hyphen, so it's bit.ly/sn569. That's a quick way of getting there because Google hasn't found it yet. Now, if you do put a hyphen in, as I usually do, instead you get taken to a Google search for "Steve Gibson smelly old man."

**Leo:** Oh, lord.

**Steve:** Because this is the Internet.

**Leo:** Somebody figured out your scheme.

**Steve:** Oh, it's a diabolical scheme, yeah. It's very difficult to anticipate...

**Leo:** Unbelievable.

**Steve:** ...what next week's bit.ly link will be. So, yes, sn-569. And I'm not, unfortunately, or I guess fortunately, not at the top of the list. Some other Steve Gibson gets the benefit of being the smelly old man.

**Leo:** What? Well, we've got to fix that. Come on, folks. We can do that. We can make that work. That's funny.

**Steve:** So this is exactly what we want. It's written from scratch by a guy who's got a nice set of software of different kinds of applications. So he says: "Automated system certificate store checking for Windows workstation and server alerts users to the addition and removal of system certificates." He writes: "This new, free utility will monitor any changes made to the Windows certificate stores on your system. Certificates can be added or removed to your system for a variety of reasons. Windows Updates, new software packages, et cetera, can make alterations to the certificate store. Unfortunately, some malicious software could also add an all-purpose certificate and essentially create an attack vector for SSL/TLS man-in-the-middle attacks or provide a foothold for a bad agent to usurp and exfiltrate information from your system without your knowledge.

CertWatch performs hourly scans of all system certificate stores and will report any additions or deletions from those stores when changes are made." So, yay. Beau, thank you. On the page he references, that's how I was able to find SN-551 so quickly, he made a little note that he got the idea from listening to that dialogue and wrote it. And many people - there are a couple people have come up with some sort of solutions more like what I was suggesting, a little, I don't want to call them a kludge, but not just a nice, clean little piece of freeware, purpose set, that just does an hourly check. We have that now. So again, thank you. And everybody, again, it's bit.ly/sn569 with no hyphen. And the product is called CertWatch, or the app is CertWatch. And again, Beau, thank you for solving this problem for us.

Okay, now. Oh, boy. Twenty-year-old designed-in Windows behavior lets printers, or anything pretending to be a printer, install malware.

**Leo:** Well, why not?

**Steve:** All the way back, starting with Windows 95.

**Leo:** Oh, lord. Which, by the way, underscores your contention that many of the pieces in this brand new modern Windows 10 are kind of old.

**Steve:** Yeah.

**Leo:** What subsystem is this?

**Steve:** Get this. Okay. So, well, and so we have a problem, Houston, that Microsoft made a change on Tuesday which allowed these guys, Vectra Networks, to go public with what they found. So they wrote in their own blog: "Security researchers with Vectra Threat Labs" - and it's VectraNetworks.com - "uncovered a critical vulnerability" - and, yeah, and it's got a CVE number, but unfortunately it's not something, well, as we'll see, that can really be fixed - "which affects all versions of Microsoft Windows all the way back to Windows 95. The vulnerability is created by the way Windows clients interact with network printers, allowing an attacker to execute code at the system level" - so full system privileges - "either over a local network or" - are you sitting down? - "over the Internet."

**Leo:** No.

**Steve:** Oh. Twenty years ago, they write - oh, no, I guess this is me. I'm sorry. Twenty years ago Microsoft implemented a very dangerous feature known as Microsoft Web Point-and-Print Protocol which allows - and think about this. I mean, we've seen this in action.

**Leo:** Oh, yes.

**Steve:** Which allows a Windows machine connecting to a network-hosted printer for the first time to receive and install a printer driver delivered from the printer. What could possibly go wrong? And I have to say I was reminded of the very similar Windows Metafile design mistake Microsoft made during the same time period.

**Leo:** Right, right.

**Steve:** Which was one of our very first podcasts. When I looked at the Metafile code, it was immediately clear to me that this was not a mistake. This was on purpose. Now, it was misinterpreted. Even my analysis was misunderstood because I never said this was malicious. People just didn't understand that, when this was done, it didn't seem like a bad idea because nobody - it was like "The Wrath of Khan," where he didn't raise his shields, and he's approaching Kirk on the Enterprise, and Kirk is saying, well, this is mighty odd, and Khan says, "We're all one big happy Federation," you know, "no need to raise shields." Similarly...

**Leo:** It was a perfect time, Steve. Kids played outside.

**Steve:** Precisely.

**Leo:** You didn't lock the doors.

**Steve:** Dogs did their business, and everyone just walked off.

**Leo:** We didn't have cell phones.

**Steve:** You avoided the landmines, yeah.

**Leo:** No answering machines.

**Steve:** Right. So back when the Metafile format was created, someone said, hey, how cool would it be if a token accepted a pointer that jumped to code in the Metafile? Well,

fast-forward 15 years. Oh, my god, you know. And so people thought I was crazy for thinking that Microsoft would have ever done this, except I will note that Mark Russinovich agreed with me, and we would respect his opinion, as well. So this is similar. This is back with Windows 95, when IPX and SPX and Novell NetWare - and Bill was still kind of thinking, you know, we've got to buy a bunch of modems at Microsoft so people can call into the Microsoft Network, and we can compete with that AOL thing and CompuServe and so forth.

And so back then, what a convenience. You could bring in a laptop, plug it into the network. Now, the network might have a shared printer. Well, it's going to need a driver. But who knows what driver? How convenient for the printer to provide it to the computer. So it's called Plug-and-Print. And of course we remember Plug and Play, or Plug and Pray, from the day. Oh, I'm sorry, it's Point-and-Print. And so what Microsoft did was design a protocol with no user interaction because that would confuse people. You just want it to work. So even in today's OS under 7 or 8 or 10, it bypasses UAC, no notification at all. It installs a kernel driver, which is, like, god power, in the kernel of the OS from anything that your computer finds that looks like a network printer. And it isn't even restrained to the LAN.

So taking it kind of calmly, more so than I am, Vectra said: "Most organizations try to apply the principle of least privilege to the devices in their networks. This works pretty well for things like laptops or desktops since the hardware they use doesn't change very often. However, printers are a bit different. While they still need drivers, printers need to support virtually any user that wants to connect to them. As end-users move through a building, they naturally want to use the printer closest to them." Because, you know, it spits out that paper that they have to then go get. So they don't want the printer on the 12th floor to be printing it when they're on the third floor.

"Mobile users expect to be able to easily connect and use a printer when they come into the office. In addition, most organizations don't standardize on a single printer, and will have multiple models and manufacturers often within a single network. So instead of having system administrators push all possible printer drivers to all workstations in the network, the solution was to develop a way to deliver the driver to a user's device right before the printer is used. And this is where Point-and-Print showed up." A happy name.

"This approach stores a shared driver on the printer or print server, and only the users of that printer receive the driver that they need. At first glance, this is a practical and simple solution to driver deployment. The user gets access to the printer driver they need without requiring an administrator - a win-win. The issue? The problem is that, for this scheme to work nicely from an end-user perspective, an exception was required. Normally, User Account Controls are in place to warn or prevent a user from installing a new driver. To make printing easier, an exception was created to avoid" that pesky - I'm adding that, editorializing - UAC control. "So in the end," they write, "we have a mechanism that allows downloading executables from a shared drive and run them as system privilege on a workstation without generating any warning on the user side. From an attacker perspective..."

**Leo:** It's amazing.

**Steve:** "...this is almost too good to be true, and of course..."

**Leo:** Vectra calls it a watering hole attack.

**Steve:** Well, yeah, exactly.

**Leo:** Which I love. I mean, I don't love the attack, but the idea is you just - all you have to do is infect the printer.

**Steve:** Right. And not only will your system get infected, but reinfected.

**Leo:** Yeah, yeah.

**Steve:** Every time it tries to print to it.

**Leo:** Put it on the printer.

**Steve:** So they said: "This is almost too good to be true, and of course we had to give it a try. Researchers at the security firm Vectra Networks" - they're speaking in the third person - "discovered that the Windows Print Spooler doesn't authenticate print drivers when installing them from remote locations." Because we're all one big happy world. "That lack of authentication makes it possible for attackers to use several different techniques" - I mean, again, this is such a gaping hole, you don't have to - it's not like worming your way through some complex incantation of four different exploits that have to interact perfectly, and only when the ASLR happens to land in the right place. No. This is by design from Windows 95 and has never gone away.

They said: "The lack of authentication makes it possible for attackers to use several different techniques that deliver maliciously modified drivers instead of the legitimate one provided by the printer maker. The exploit effectively turns printers, printer servers, or potentially any network-connected device masquerading as a printer" - and remember, that's just a protocol thing. So, yes, a light bulb from China could pretend to be a printer. And it's like, oh, I didn't know there was a printer there. Let's send some documents to it. And it takes over your machine.

**Leo:** Unbelievable.

**Steve:** "Into," they write, "into an internal drive-by exploit kit that infects machines whenever they connect."

**Leo:** I just want to say, when you come to this studio, if you ever want to use our printer, please, be my guest. Just go right ahead.

**Steve:** You'll walk away with a free gift.

**Leo:** A free gift.

**Steve:** And then I wrote...

**Leo:** That's been in there for 20 years.

**Steve:** Yes. But wait, there's more.

**Leo:** Oh, no. More?

**Steve:** Vectra Networks wrote - and I'm paraphrasing from what they said for a bit more emphasis. Under "Infecting Remotely Using Internal Printing Protocol and webPointNPrint," they write: "So far we have constrained ourselves to an internal network where a device was either inserted or infected and used to further infect devices connected to it." I mean, so understand, they've done this. And their blog posting has, chapter and verse, the whole - it's all laid out. "Internet Printing Protocol (IPP) and webPointNprint allow us to extend this issue outside the Intranet to the Internet. IPP [Internet Printing Protocol] allows for the same mechanism to load drivers from remote, in this case very remote, printers. This can be done with the following piece of code from the MS print server." And then their blog shows it.

Now, this was "fixed," in quotes, last Tuesday. What did Microsoft change? Well, they're unable to change this behavior without crucially and critically breaking 20 years' of roaming laptop and mobile computing transparent printer driver installation. So they added a dialogue. And we've seen how well those work with, for example, not upgrading to Windows 10. So the good news is security-conscious people can disable Point-and-Print. As I dug into this, I didn't have a chance to go any further. I will probably see what it takes to do that and provide some guidance for next week because I imagine our listeners will be very interested in perhaps not having this happen to their users. You can push it through group policy, so a corporation could do this. And so essentially the only thing Microsoft could do is make it less transparent. They could not break it because too much of the existing infrastructure depends on this behavior.

But this is, again, this is a classic example of something that back in 1995, with Windows 95, seemed like a good thing. And then, now, someone did deliberately extend it to the Internet Printing Protocol. It might have been wise to say, you know, let's not let remote printers on the Internet install kernel drivers without any user interaction. Someone should have said no to that because that was in more recent times. But it's probably still, who knows, maybe 15 years ago. So, unbelievable. Yes?

**Leo:** Well, the good news is printers always auto update. So you don't have to ever - never mind. We continue on with the security news of the day.

**Steve:** So this is a quickie, just a note. I don't know if this will affect any of our users. But a startling flaw was found in a major vendor's router. I guess arguably Cisco is the granddaddy of Internet routers. Juniper, however, has a great reputation for high-end big-iron routers that form the backbone of the Internet. So they released a security

bulletin after somebody found a little bit of a problem with their certificate management system. It affects every Juniper router that uses the so-called Junos OS. That's the operating system that spans their product line.

And under "Problem" the report said: "Junos OS runs PKId" - so that probably means Public Key Infrastructure daemon, a background service - "for certificate validation. When a peer device presents a self-signed certificate as its end entity certificate, with its issuer name matching one of the valid CA certificates enrolled in Junos" - okay, so that means that a self-signed certificate is being looked at, not is being trusted, not based on itself, but on the name in the certificate, matching the name of a trusted certificate. And since it's self-signed, anybody can create that certificate using whatever names they want in it because they're making their own certificate and signing it.

So they don't have to convince anybody else to trust them and look at the contents of the certificate and validate it and verify it. No, so they're able to use a trusted name on a certificate they make, self-sign it, and across the entire spectrum of Junos devices, "The peer certificate validation is skipped, and the certificate is treated as valid." And they say, "This may allow an attacker to generate a specially crafted self-signed certificate and bypass certificate validation." And that's not good.

So if by any chance any of our listeners is responsible for or has Juniper routers within their organization, I'm hoping that the news of this was spread through whatever contact system Juniper has. But this is something that could quickly be exploited, the idea being that this is how the routers are essentially peering. And the last thing you want to do is to allow a malicious agent to peer with a major router because, as we know from all the past problems, the inadvertent mistakes with BGP routing, even if they're mistakes, horrible things can happen. And if it's malicious, then all bets are off. So this is something - I'm glad they found this. There was no reported known abuse. So this was someone detected this and said, Juniper, you might want to take a look at the way you're handling certificates.

And speaking of Juniper and Cisco, it is Errata time. Our friend David Redekop was first, and several others corrected my mistake last week when I said that the Cisco SG300 managed switches ran IOS. It turns out it is an IOS command line interpreter, but not actually the IOS firmware itself. So I appreciated the correction, and I wanted to correct the record.

**Leo:** You sent me a note saying that you liked this $50 EdgeRouter from Ubiquiti.

**Steve:** Yo, and that's where we're headed next. First item in Miscellany.

**Leo:** Okay.

**Steve:** So you could have three dumb routers, or one super smart router. I wanted to put this on everybody's radar. We talked last week about the concept of using a managed switch. Well, a managed switch is just that. It's like an unmanaged switch; but you're able, for example, because it has additional processing power, do things like filter packets coming in and out of specific ports on the switch. But it's not itself a router. And I actually stumbled on this, and then I tweeted it, and a lot of our listeners already know. And it was a 100% rave response from existing owners who listen to this podcast or, rather, who follow me on Twitter, which is probably a subset of podcast listeners. And I

just stumbled on it. I was looking at something on Amazon, and it was people who bought this also looked at this. And I said, ooh, what's that? Okay, so get this. Well, in fact, here it is.

**Leo:** I'm going to get it, yeah.

**Steve:** Oh, and Leo...

**Leo:** Is this it, the six-port router? Or is it the...

**Steve:** Look at this. It is the cutest little thing.

**Leo:** Yeah, that's it. All right.

**Steve:** I mean, it is just itty-bitty. And it has the highest Packet Sex Density of any little box I've seen. A PSD, that's a technical term for what this thing does [Power Spectral Density]. So, yeah. So for those who can't see the video, it is a cute little fanless, and I should say $49, amazingly capable smart router. Now, what's different about this compared to any of the routers we talk about is that this has five logical interfaces, not just five physical interfaces.

So, for example, the normal kind of router that we're used to talking about, what I call the "blue box" routers, like all the Netgears and everything else that people use, architecturally that's a two-interface NAT hub, essentially, sort of a two-interface NAT box with a WAN side and a LAN side, connected to some number, typically 5- to 8-port switch, meaning that all it really is, is sort of a dumb switch with a router behind it. What's different about this is that you actually have five logical interfaces. So the point is, for $50 - and by the way, this is called the Ubiquiti EdgeRouter X.

**Leo:** Now, is this the one with SFP or not? And what the hell is SFP? Do you know what that is?

**Steve:** SFP or POE?

**Leo:** There's POE. So they have two. There's one with two POEs and three regular Ethernets; and then there's one with one POE, and the other one is SFP. Oh, SFP is fiber, okay.

**Steve:** Yes, correct.

**Leo:** So I don't need the one with fiber.

**Steve:** No. You don't want the one with fiber. So what is beautiful about this is, I mean,

this is the answer. What it doesn't have is wireless. So if somebody wanted truly a single-box solution, we don't have that yet.

**Leo:** But Ubiquiti does have really great WiFi routers...

**Steve:** Yes.

**Leo:** ...that work with it; right?

**Steve:** Yes. Yes. And in fact that's probably why these routers exist. They call them "edge routers" because then their stuff is able to plug in and use this. So I just want to make sure that people understand that this one little cute $49, $50 box does the whole network isolation thing that we need for IoT. Each one of those interfaces can be its own subnet, with inter-subnet isolation so that one cannot see the other, yet they have managed access to the Internet. So you could, for example, if you wanted to use this and turn your existing WiFi router - most WiFi routers, you can reconfigure them to be just a hotspot, and so plug that into this in order to get WiFi.

Anyway, this is so much less expensive and so much more powerful than that SG300 Cisco that I was talking about last week, that I want to make sure people know about it. It's just - it's a beautiful little box. Now, there's one limitation, the only one I know of, and that is that it is a gigabit - it's a 5Gb interface, so 10, 100, and a gig. But the processing power won't allow you to run, that is, between ports or move packets at a gig. It's estimated at about 500Mb. So as long as what you're doing, for example, your connection to the Internet is less than about 500Mb, about half a gig, then this'll do everything you want. And you could certainly connect it to a simple gig switch as your hub of a full speed gig network, where then this manages the separation of separate subnets.

**Leo:** So you could have separate WiFi subnets if you bought a couple of the Ubiquiti devices.

**Steve:** Yes, yes.

**Leo:** And have them completely isolated as in the three-router solution.

**Steve:** Yes. I mean, and so it's like either three dumb routers or one super smart router.

**Leo:** Nice.

**Steve:** And so this little thing, it's just a little gem, for 50 bucks. I just, like, on my...

**Leo:** So this sets up your IoT.

**Steve:** Yes.

**Leo:** If you're going to do IoT, you need to get this and a couple of WiFi access points.

**Steve:** Now, it does not support OpenVPN, but it does support both point-to-point tunneling protocol, PPTP, and IPSec. And so that means you can also create - and it does that normally for intersite linkage. So, for example, if you had two facilities, you could establish a static VPN link between two of these in different locations and bridge the networks together. Or you can run this with PPTP, which is a widely supported VPN protocol, the mobile devices support, for example, and have access to your home network securely over point-to-point tunneling protocol.

So, and of course, remember, as we know, it's easy to run OpenVPN on a Raspberry Pi. You just hook it up and give a bash script command, and bang, you've got OpenVPN installed on a little tiny Raspberry Pi. So if you wanted OpenVPN, the Raspberry Pi could plug into one of these ports and then offer its services, OpenVPN-compatible connectivity, for you when you're roaming outside. Anyway, we're beginning to assemble a very nice little toolkit of pieces to build solutions. And this is the kind of thing I know our listeners are interested in, so I wanted to make sure everybody knows.

I finished Peter Hamilton's "The Endless North Road" novel ["The Great North Road"]. I really did enjoy it. But it was quite long. And I'm now waiting for the sequel to "Beyond the Abyss of Dreams" or whatever that was called ["Abyss Beyond Dreams"], and also for the next of the Rho Trilogy, "The Altreian Nexus" or something series. So I'm going to go back - I never finished, actually Jack Campbell is the author of a series I've talked about before, "The Lost Fleet" series. And I really - I love the concept, and the writing was enjoyable.

Our listeners who have been listening for a long time will remember that - real quickly, the premise is that in the far future a commander is found in a life support capsule after 100 years of battle. And due to the aggression, the attrition rate of command officers and ships is very high. Training standards have fallen. And they end up reviving him, and he's sort of legendary because of the way he died. No one could believe he's still around.

The point is he has knowledge of space combat which was common then and has been completely lost, just due to the fact that commanding officers are getting killed too quickly, and people are being promoted up the ranks. And the people who remember me talking about it before will recall that the way he paints these battles, if you like space opera, you are just going to be so happy. They're just clever and wonderful things he sets up, an interesting universe, and a whole bunch of books. I think there were, like, six in the "Lost Fleet" series, and there's now four in what he calls "Beyond the Frontier."

And so I started to - I thought, okay, I'm just going to read the last couple chapters of the last book of the first series of six to kind of remind myself who the people were and so forth, to sort of bridge me in because I never took the leap into the "Beyond the Frontier." And I did that for maybe 10 minutes, and I said, okay, I've just got to start from scratch. So I'm starting over because they're just so enjoyable. So if you haven't read them, you like well-written space combat, it's always been on my recommended sci-fi reading list. And so I'm back, and I'm going to re-read the ones I read years ago and then move forward, patiently waiting for Peter F. Hamilton and Richard Phillips to both get their next books finished.

And I mentioned at the top of the show, Star Trek's 50th Anniversary year. That is, this is 2016. Star Trek was born in 1966. Netflix has just announced that they, well, actually, as Popular Science's coverage put it, "Netflix users worldwide will be able to boldly binge where few have ever binged before."

**Leo:** Oh, please.

**Steve:** By the end of the year, 727 episodes of all previous Star Trek series will be available worldwide on Netflix for streaming. And CBS's much-anticipated new Star Trek series, beginning in January of next year, 2017, will make new episodes available for streaming on Netflix within 24 hours of their debut, except for viewers in the U.S.A. and Canada. U.S. viewers will need to go through CBS and Canadian viewers through Crackle until...

**Leo:** Is it Crackle? Somebody in the chatroom says it's Crave TV.

**Steve:** Okay. "Crackle" is what Popular Science wrote. So I don't know.

**Leo:** That's Sony's network, Crackle, yeah.

**Steve:** Ah. So anyway, and I never - I kind of faded out on "Deep Space Nine." I watched the first couple seasons. Mark Thompson absolutely promises me that it got really good, that like that whole Cardassians and some dimensional creatures I never - I didn't really pay attention to it. But every other series I watched and really enjoyed. So, hey, and there it is.

**Leo:** A special Time magazine.

**Steve:** Wow, the 50-Year Anniversary issue.

**Leo:** "50 Years of the Final Frontier." "The most influential science-fiction series ever."

**Steve:** I mean, they're just iconic, Leo. And it was only three seasons.

**Leo:** That's the thing that's amazing. It was really a failed show. Or not failed, but it wasn't super successful until...

**Steve:** Well, you know why, because at the time - I've told this story on the podcast before. But I had dinner with Gene Roddenberry during a Comdex.

**Leo:** Wow, that's cool.

**Steve:** And I told him, you know, I didn't want to drool all over him, but I told him that I was a serious Star Trek fan. I don't think I mentioned that my port scanner was called ShieldsUP!.

**Leo:** Wait a minute. How long have I known you? Duh. ShieldsUP!. I never put two and two together.

**Steve:** Yup. So, but anyway, he explained that back then Nielsen had not yet implemented demographics. And the raw count was all they were using. And so at the end of its third year, though it had a huge fan base, the numbers weren't there. And so it got canceled by Paramount, who said, eh, sorry, Kirk. Maybe you can be an attorney on some comedy show in 50 years.

What happened then was that the Nielsen system began to work on - they understood that - I guess maybe they had the processing power, or they just wanted to do an upgrade. They decided maybe who is watching matters. So they had in their archive never-processed demographics. And in building their statistical model, they went back and reprocessed data they had collected but never used. And it turns out that Star Trek had the most perfect demographic profile of any show in the history of television. You couldn't design a show that had newlyweds buying homes and cars and diapers and soap and everything that advertisers want, I mean, it was like, it was the young newlywed yuppies, and they didn't know it. So they canceled the series. When in fact it was doing fabulously well with that particular demographic. So that was directly from Roddenberry's mouth.

**Leo:** I believe it, too.

**Steve:** And a great author. So I just did want to follow up on last week's "Mr. Robot." And I want to say I hope we soon spend less time inside of Elliot's disturbed head and get back to some fun hacking.

**Leo:** Really.

**Steve:** And bring down our evil corporate overlords.

**Leo:** Everybody was raving. I haven't seen it yet. Everybody was raving about it around here.

**Steve:** Okay. Well, I'm - it's like, whoa. It was a little dark and kind of sad.

**Leo:** Well, everybody who works for me is dark and sad, so that's probably why.

**Steve:** I'll watch it, but, whoa. And I got a tweet from Michael Cunningham who said, who asked me, "Did you notice the QR code in Elliot's journal actually works and goes to some goofy site?"

**Leo:** Oh, there's lots of Easter eggs in this show; right?

**Steve:** Yes. ConfecturaIndustries.com. And I put the link in the show notes, and The Verge covered it. It has a huge blown-up QR code.

**Leo:** I love this. It's a Geocities site.

**Steve:** It's just a weird site.

**Leo:** There's a guy with a jackhammer, under construction, under construction. Warning, this site is under construction. It's the worst.

**Steve:** One of those old-school spinning email A's from like the early days of...

**Leo:** I used to have the mailbox that opened up and the...

**Steve:** Yup, I had that, too, yeah.

**Leo:** Geocities Cool Page of the Day. Link Exchange, promote your site for free. The counter. Pretty good counter, 61,114 visitors. Wow. This takes you back, doesn't it?

**Steve:** So I have a puzzle. I know that the puzzles that I like get a lot of feedback from our listeners.

**Leo:** I'm still stymied by the last one, by the way. I'm, like stuck on Level 5 or something.

**Steve:** Of The Sequence?

**Leo:** Yeah, well, not Level 5, maybe Level 10. Yeah. Because, well, there was one, and it's like, oh, the things can move another thing. So then...

**Steve:** Ah, yes.

**Leo:** ...that's the one I got stuck on for a while. And then I figured that out. Now I'm

on the one after that, and I have no idea.

**Steve:** I'm at, like, 49 or something. I think I'm on 48, two away from finishing that big board.

**Leo:** "You're a better man than I am, Gunga Din."

**Steve:** And I'm like, my attention just sort of wandered. Okay, now, everybody loved Hook.

**Leo:** Yes.

**Steve:** Remember Hook.

**Leo:** Yes.

**Steve:** Which was that black-and-white puzzle where you tap little things, and it pulls these little rods back, and you have to do it in the right sequence in order to sort of unhook everything. And we loved it, and we were uniformly disappointed when we solved the first 50 levels, and there was no second 50 levels. And so I contacted Hook's author, and I can't pronounce - M-A-C-I-E-J.

**Leo:** Yeah, it's a Polish name.

**Steve:** Can you help me out, Leo?

**Leo:** I think it's pronounced Sharday.

**Steve:** No. Really?

**Leo:** I'm kidding. I don't know.

**Steve:** Okay. M-A-C-I-E-J - I'm sorry I don't know how to pronounce your name - Targoni. Anyway, I wrote to him with our disappointment that Hook ran out. And I suggested - oh, and I told him, I said: "This is why I think the game is perfect." And I ran through my theory of like the perfect puzzle toy thing. And he replied, he said: "I'm thinking constantly about expanding it, but I think I" - and we're talking about Hook now, and this is months ago - "but I think I will just go with a new game. The main design idea was to add new game mechanics continually, to maximize a-ha moments."

**Leo:** Ooh. That's clever. By the way, here's the pronunciation. It's from PronounceNames.com: "Ma-chee."

**Steve:** "Ma-chay."

**Leo:** "Ma-chay."

**Steve:** Right. "Ma-chay," good. Thank you, Leo. So he said: "At this stage there is not that much interesting things to add. And I don't want to stretch play time just for the sake of it. Game has to be fun from the beginning to the end."

And so I replied: "There's another way to look at it, though, which I will try to explain. Once someone has acquired a new skill or understanding, it can be fun to simply use that new skill for at least a while, even if nothing further is being learned. It can just be a pleasant diversion to solve the puzzle, even if nothing more is being learned about the puzzle. Just working the machine and accomplishing is enough."

And I wrote: "Perhaps the best example is the enduring popularity of crossword puzzles or Sudoku. The people who enjoy them are perhaps getting a bit better at them over time, but the pleasure is just in using their brain to work out this particular solution, which always changes." And of course we've also run across Infinite Loop, which is a lot of fun, and Infinite. So it was that notion. And so what I proposed to him even before Infinite Loop was discovered, I said I wondered "whether it would be possible to create a Hook level generator which is capable of endlessly creating Hook levels by itself. That could be a Hook Pro or Hook Ultimate, which people could play and play and play, just for the pure pleasure of solving different Hook puzzles."

And finally he replied: "It's my second 'serious' game, so I think I can do better." He says: "I will stay within puzzle genre and keep exploring it, looking for unique ideas. And my design will keep on going in the direction you are calling a 'sweet spot' slow progress, relaxing, no timers, no rush, no three-star reward system."

**Leo:** Steve, you actually have people designing games for you now. To your criteria.

**Steve:** No, but I'm trying to encourage the world to produce more puzzles that we like. So this is a dollar. It's 99 cents, available for iOS and Android and Steam. Now, Leo, if something's available for Steam, what does that mean?

**Leo:** It's on PCs.

**Steve:** Okay, cool.

**Leo:** And actually, oftentimes, if it's Steam, it's also on Macs and Linux. It just depends if the developer makes it portable. But Steam is a game store on PCs, basically.

**Steve:** So let me be clear. He has done it again. And it is so - it's just right.

**Leo:** Good.

**Steve:** There's no instructions.

**Leo:** Nope.

**Steve:** He just presents you with something, and you go, okay, and you start poking at it. And then you begin to figure out how it works and what it does. And it's funny reading what he wrote because he described his intention, which is here, because this starts out simple, but then it evolves. It adds new features. And the one with the black dots threw me for a while. I was solving them, but I didn't know why. Now I understand what they're about. And so, and I've not finished it. So, and I've gone deep. So I don't think this thing is going to exhaust itself and disappoint people too quickly. So top recommendation, from the original author of Hook, for a dollar on iOS, Android, and Steam. Oh, and it's called Klocki, K-L-O-C-K-I. I've got links in the show notes. The website is KlockiGame.com. K-L-O-C-K-I-G-A-M-E dot com. Klocki, K-L-O-C-K-I. I think everyone's going to have a lot of fun with it.

I wanted to quickly mention, we've been having great success with the evolving Healthy Sleep Formula. The number one reported side effect, aside from sleep, which is the intended goal, is exactly what you experienced the first time you tried it, Leo.

**Leo:** Yeah, yeah. I've had it happen again.

**Steve:** Headaches and groggy hangover in the morning.

**Leo:** Yeah, yeah.

**Steve:** The solution is to cut the dose in half.

**Leo:** Okay.

**Steve:** That big monster pill, that 1,500-milligram niacinamide, it turns out, if you just bring a sharp edge down in the center of it, it just breaks into two pieces. And it's time-release, but it's time-release mass. So cutting it in half doesn't destroy the time release. I've run it at half dose. Works perfectly. And everybody who reported a headache and grogginess in the morning had it resolved and no more grogginess.

**Leo:** Okay.

**Steve:** So it was just for some people who are more sensitive, the whole pill is too much,

and the half pill - and so what I was curious about was whether half of it would relieve the side effects that are negative while preserving the function, and it does. Now, here's the bad news.

**Leo:** There's no niacinamide to be found anywhere in the United States of America now, thanks to you.

**Steve:** There isn't.

**Leo:** Really?

**Steve:** There isn't.

**Leo:** You've got to start a side business.

**Steve:** I kept adding links: Amazon, then iHerb. Then they sold out. Then I added Swanson. They sold out. I added Vitamin Shoppe. They sold out. I added Drugstore.com. They sold out. I added eVitamins. They sold out. So Amazon, iHerb, Swanson, Vitamin Shoppe, Drugstore.com, and eVitamins. Nobody has it. I was exchanging tweets last night with somebody who spoke to Source Naturals, and they said, "We don't know what has happened. But something has happened."

**Leo:** I would imagine that the demand for niacinamide has been very consistent over the last 50 years.

**Steve:** And low.

**Leo:** And all of a sudden, you know, "We've been making 150 pills a day for as long as I can remember." And then suddenly, "We sell that many a minute now." Of course they don't. It's like, what happened?

**Steve:** Yeah.

**Leo:** But I'm actually thrilled to hear that because, if I cut it in half, then it's really not even - what is it, then, 250? How big is that pill? It's a horse pill.

**Steve:** It's 1,500, so it's 750.

**Leo:** 750. And then one milligram, one milligram of melatonin time-release.

**Steve:** Yes.

**Leo:** Good, I'll try it. Now it's even less and more, you know. You know.

**Steve:** And it's far away from any concern, yes. And there isn't any. So anyway, so when I get a chance, I'm going to talk to Source Naturals. He said that they said they're not going to have any until the - I think he said the end of September.

**Leo:** Chinese niacinamide shipments.

**Steve:** Well, and that's the other problem, is there are only two time-release niacinamides. The other one never worked for me. So it's probably not very delayed. And so this is, unfortunately, this Source Naturals is the only one I know. Now, what I would love to do would be to have them cut it to maybe a thousand milligrams and add a milligram of melatonin. Then they would have a single beautiful little time-release happy night pill. So anyway...

**Leo:** That's the name of this show, by the way, "Steve's Happy Night Pill," available in pharmacies.

**Steve:** And Leo, 2,000 people a day now, every single day - it was at 3,500 for a while - 2,000 people a day go to that page.

**Leo:** Wow.

**Steve:** And they just sort of take it for granted, "Oh, yeah, now I sleep perfectly. You've changed my life," blah blah blah. So I appreciate it.

**Leo:** Well, I'm saving it for my trip to Paris because jet lag's a bitch. And so I'm looking forward to having something that can help me with that. Anything that can help me with jet lag is going to be...

**Steve:** I've found yet another way of telling our listeners about SpinRite.

**Leo:** Uh-huh.

**Steve:** Not that anyone who's been listening for long doesn't know. But yesterday at 9:26 a.m. Emett Speer sent me a tweet. He said - and this was the second of two. He said: "I also wanted to add that I am an owner of SpinRite 6. I was trying to get my company to purchase a corporate SpinRite 6 license by purchasing four, for use on the many PCs they own. I demonstrated its abilities by using it to recover a dead RAID 6 array for a customer who was in a panic to get their data back. The company and customer were grateful that I was able to recover the full system with no data lost, but they didn't get SpinRite for our PCs."

To which I say, well, win some, you lose some. I'm happy to have performed a service. They now know that it works. In some neuron in their head they've stored that information. So next time the CEO's machine dies, just after pulling all of his financial data together, and he doesn't have a backup of it yet, Emett can say, you know, go over here and buy a copy. So Emett, I appreciate you exposing them. And, boy, for a RAID 6 array to need SpinRite, that's negligence because in a RAID 6 you have two drives of full redundancy. Any two drives can fail, and you're still okay.

**Leo:** And apparently they did.

**Steve:** So three drives had to fail in order for this to be a problem. So the flipside is that makes SpinRite's job much easier because it only needs to recover one of the three drives.

**Leo:** Right, right.

**Steve:** In order to bring the RAID back. And then you can do a rebuild of any that it can't recover. So SpinRite in the era of RAID still makes sense. And again, Emett, thank you for providing that news.

**Leo:** And actually SpinRite would still work on a ZFS disk on FreeBSD. You just have to boot it in FreeDOS.

**Steve:** Yup.

**Leo:** Because it doesn't care what the file system is. It's not looking at that.

**Steve:** Yup, exactly. So we talked last week about Facebook's addition of secure end-to-end encryption as an optional feature to Facebook Messenger. And they call that Secure Conversations. So you're able to essentially promote an interaction with Messenger to a Secure Conversation, sacrificing Facebook's server monitoring of the dialogue, which means that some features you may use for standard messaging would no longer be available. But that's the tradeoff with essentially putting your communications in a secure tunnel. And last week I mentioned that the whitepaper explained some additional features that we ran out of time - we had a full two-hour podcast last week. And so I thought, okay, let's - I want to get into this and look at it and understand it.

So abuse reporting and secure storage management they detailed. And those of our listeners who enjoy crypto puzzles will like this. Facebook has layered on top of the Signal protocol what they call "franking." So the idea is they're concerned about abuse of service, that is, abuse of their terms of service, in encrypted conversations, because if it's encrypted end-to-end, they can't see them. They're not able to perform any kind of filtering, for example. So they had to come up with a way, without violating privacy, to allow the recipient of a message which they found objectionable to be able to report it in a fashion that was cryptographically sound. Which would mean, of course, first of all, they wouldn't see the communication until and unless it was reported. And if it was reported, then they needed the property of nonrepudiation, that is, they needed a

system whereby the sender was unable to say, "I never sent that. The person receiving it made that up. That didn't happen."

So, "Any participant in a secret conversation may voluntarily notify Facebook of abuse of content they receive." Technically they could notify Facebook of abuse of content they send, but I don't think that seems likely. "Facebook uses such reports to identify users who violate," writes Facebook, their "terms of service. The ability to report abuse does not relax the privacy guarantees inherent in the endtoend encryption of Secret Conversations. Facebook will never have access to plaintext messages unless one participant in a secret conversation voluntarily reports the conversation."

Okay. So here's how this works. So we have the sender of any message must incorporate what they call a "franking tag," which is appended to and then encrypted with the message. And the franking tag actually is just a cryptographic signature that we've talked about often. In this case, the sender on a per-message-sent basis creates a nonce, a 256-bit one-time random number. They use that as the key to an HMAC. Remember that an HMAC is a hashed message authentication function, in this case an HMAC 256. So they have their conversation. I'm sorry, they have their message. They make a random number, and they first stick it on the end of the message. Then they use that random number as the key to hash the message, which generates a signature. That signature is the franking tag. And then, after getting the franking tag, they destroy that random 256-bit nonce. Don't need it anymore. So it's at the end of the message, and it is the key to the keyed hash that generates the signature, which is the franking tag.

So the sender of any message must incorporate that franking tag into the message and send it along - oh, and then they use their own keying, their existing cryptographic key in order to encrypt both the message and the franking tag as one. They then send to Facebook, because it's going to go to Facebook and then on to its destination, they send to Facebook the encrypted message and the franking tag. Now, no recipient of a message will display any message without a valid franking tag, which is validated upon receipt. So that essentially - so if someone received a message where the franking tag did not validate, the app doesn't show it. So there's no way to spoof or change the message in a way that the franking tag would not detect.

So the recipient would decrypt the message using, again, the same secret key that they had previously negotiated. Then at the end of the message would be the nonce that the recipient generated. So they would take that and validate the franking tag by using that to key the same HMAC function feeding the same plaintext plus that nonce through the HMAC to get the franking tag. So that's at the far end. Now, in the middle, Facebook receives from the sender that message plus the franking tag. Facebook then uses their own secret key, which they keep secret, private, to key another HMAC where they concatenate the franking tag with what they call the "conversation context," which is the sender and recipient identifiers and the timestamp. And this creates the reporting tag.

So Facebook then sends that reporting tag, the franking tag, and the encrypted message blob that they could not see into, all to the recipient. The recipient, as I said, verifies the franking tag and, if it's verified, will display the message. Now, if the recipient finds the message objectionable, they're able to, and Facebook said they're still working on the UI for this phase, and so they don't have it yet. But the recipient can push a button to report the sender as having sent this. So what that button will do is return the reporting tag, the franking tag, and the plaintext which the recipient has seen and found objectionable, all to Facebook.

What that then allows is for Facebook to perform all the verification, because it will have the plaintext of the message. That'll contain the nonce that it never saw before, and the

franking tag. And the franking tag was bound into the reporting tag, which is HMAC'd under Facebook's own private key. So Facebook again uses their private key to validate that nothing has changed, and that allows them to absolutely cryptographically verify that the sender sent what the recipient claims because their identities are bound in; also that the sender sent it when the recipient said because there's a timestamp bound in. And you can't change a bit anywhere, or the whole thing refuses to go.

So there is the abuse reporting mechanism which Facebook added to the secure communications in a way that doesn't violate security, but creates non-reputability of a message received that somebody finds objectionable. And only if a message is reported does Facebook ever see the plaintext. And that comes from the decrypted recipient who says, I don't like what this person just sent me. So they nailed the crypto.

**Leo:** That's good.

**Steve:** It looks like they got it exactly right.

**Leo:** That's really - and so significant that they're effectively bringing this to a billion and a half people.

**Steve:** Yes.

**Leo:** That's just incredible.

**Steve:** Yes. I should note that WhatsApp is dark again.

**Leo:** In Brazil, yeah.

**Steve:** In Brazil.

**Leo:** Just in time for the Olympics.

**Steve:** Yeah. And this, apparently, I don't know how long this will last, but a judge has ordered the service suspended until further notice. Essentially, until WhatsApp complies with a court order. Which is impossible.

**Leo:** What's awesome is Telegraph and Facebook Messenger and all sorts of other messenger programs, I hope Secret and Threema, too, are getting a lot of traction because of it. This is one of the programs that almost everybody in Brazil uses.

**Steve:** Yeah. Another aspect of Messenger is called Secret Conversation Secure Storage. So, okay. So now we have Messenger, and we're interacting with a user, and we're wanting secret conversations. But clearly we need them protected on our mobile device,

whatever that is. That is, they need to be encrypted. Yet Facebook also wants the expected features to work.

So Secret Conversations, the plaintext messages, after decryption are stored permanently only on the device that participates in each conversation. And of course it does Facebook no good to store a blob in their server because they can't decrypt it. So it moves through. So it's decrypted at the receiving end. And of course there's also, in order to have a two-sided conversation log, you're storing what you sent and then what you received in response in a chain. Plaintext messages are protected using on-device symmetric key encryption. And there's an optional disappearing messages functionality which, okay, those are always kind of flaky because it's technically impossible to enforce that, as we know. But it's there because people want it. It's kind of, I guess, they think it's a cool feature.

On-device encryption ensures that messages stored permanently on a particular device are only accessible while a user is authenticated to Facebook. So in the design of this, Facebook's operational requirements were that Messenger would allow users to switch Facebook accounts. While a second user is logged into a particular device, messages of the first user are not accessible. But when the first user returns to the same device, they will find their messages reconstituted and available.

So to achieve those requirements, the clients on the endpoints employ two encryption keys, a local key and a remote key. And both of those keys are used for AES-GCM encryption. That's the right way to do encryption. It's what SQRL uses for its identity encryption because it is a simultaneous encryption and authentication. And if you don't authenticate, you don't get any results out of decryption. So it's a nice hybrid. We talked about do you encrypt, then authenticate, or authenticate, then encrypt? Because if you need to do both, the GCM cipher does both at the same time.

So the local key is generated on the device and never leaves the device it was generated on. It is used to encrypt the plaintext messages before they're stored permanently on the device. So that local key, as messages, as the conversation chain is being stored in nonvolatile memory, it runs through this local key in order to perform AES-GCM, which is a symmetric cipher. The remote key is a long-term, per-user specific key held at Facebook and delivered on the fly to the device when a user authenticates. Which is really interesting. That means that, if you're not logged into Facebook, if you're not authenticated to Facebook, those secret conversations are encrypted, and they're not visible. You get access to them by logging onto Facebook.

Facebook then sends your specific per-user remote key to the device. It's used to decrypt the local key, which you are then able to use to access your conversations. And of course when you log out the remote key is removed, and the local key is wiped, that is, the plaintext version. So you only have your encrypted copy, the local key, ready to be decrypted by the remote key the next time your particular Facebook account is logged into on the device. So, again, clean, simple, not lots of bells and whistles, and as good a secure storage solution as you could ask for.

Where there are additional hooks available in the platform, for example, iOS has a bunch of goodies, Messenger on iOS absolutely uses the various - I can't remember, there's a whole series of tags, basically property tags that you're able to give things. So you're able to say, like, tag that database as never decrypt when the user's not logged on tag. And that's enforced by the operating system in addition to the Facebook app. So it's taking advantage of the platform's features when they're there, and doing a useful job of protecting the user so that their secret conversations are not available when they're not logged into Facebook.

And then lastly, the disappearing messages is nothing fancy. It just adds a timestamp to the messages; and, when the timestamp expires, the message will no longer display. And as I said, eh, it's like, okay, I guess it's kind of a fun feature; but again, not cryptographically secure by any means because it's just checking to see if it should display it or not based on a timestamp. There's no actual way that we have yet of robustly, proactively, retroactively, I guess, removing a message that was transferred and decrypted onto a device. You just can't make it go away unless it's going to remove it itself.

Oh, and after the time limit expires, there's a grace period for reporting abuse. Then it is actually deleted. So first it's hidden. And then, to give you a little bit of time to report abuse if you choose to, it remains. And then it disappears from the storage completely.

CryptoDrop. I sounded a little negative about it last week. And based on the results they report, where do I get my copy? This is the heuristic system watcher which attempts to see ransomware scrambling files on the fly and stop it. And it performs much better, when I read the whitepaper and got into the details, it performs much better than I would have guessed. They first analyzed hundreds, literally hundreds of cryptoware samples. They looked at exactly how the files are handled. Is it a copy to a spare location and then encrypt and overwrite? Is it encryption in place? Is it a secure delete after a file move? I mean, because there are different ways you could achieve this.

So they looked at exactly what was going on. Then they found what they called three primary indicators which were suited to detect malicious file changes. One they call "file type changes." And as we know, anyone who's looked, for example, at a hex display of a file, like a DOC file or a ZIP file or an executable, there is some clearly identifiable stuff, typically at the beginning of a file, that sort of tells the operating system about what the file contains.

Now, a simple text file doesn't have that. It's just text. But anything that's sort of higher level, like a ZIP or an EXE or a DOC, that has some structure to it, that you can always see. And there's a well-known utility called "file" which determines the file type without looking, for example, at the file extension. Not all operating systems support extensions. And extensions are sometimes deliberately changed in order to confuse things. So it's got, essentially, a magic database containing hundreds of file type signatures ranging from specific programs like Microsoft Word 2007, or just unicode text, and everything in between. Certainly you could detect zip files and EXEs and DOCs and so forth. So the looking at the header, the beginning of a file, is one way to determine what kind it is.

And of course something that's going to encrypt it is going to turn that into pseudorandom noise. So suddenly the file changes its type. And that's not something you would expect to happen normally. Then they have something called the "similarity measurement," which uses something called a "similarity-preserving hash function," called "sdhash." There's a page on the 'Net that describes it, and a lot of work has been done. It kind of put me in mind a little bit of my longest repeated string concept, although this is used to do sort of fuzzy matching, whereas mine is exact matching.

This is used with overlapping hashes to do fuzzy matching of files and to preserve a set of hashes from a file in one state, or a file at one time. And then you can use that digest, those hashes, against the sdhash function in the future and get actually a number from zero to 100 of how similar the original file and the current file is. So clearly that's useful because, again, anything that's going to encrypt a file is going to turn it from something structured into noise, pseudorandom noise.

And, not surprisingly, the third indicator is the Shannon entropy, which is the

mathematical description of how random the data is. As we know, text files have relatively low entropy. But two things you can do change that. If you compress it, compression inherently removes entropy from the file because, if there is any entropy left, then there's something more you can compress there. And the other thing is encryption. Encryption turns a file into noise, which is super high entropy.

So determining the mathematical, analyzing a piece of the file for its Shannon entropy would give them a quick sense of whether this looks like it's encrypted gibberish or something that has lower entropy and is probably information that something can understand. So armed with that, they did a bunch of lab testing. They obtained 2,663 malware samples from VirusTotal, using ransomware-related search terms and known ransomware virus variant names, and essentially ran through - there were many that had tiny variations among them. So there weren't, like, 2,663 completely different types of ransomware. We know that's not the case. But it's versions of TeslaCrypt and CryptoLocker and so forth. And so they separated that out. And they ended up settling down to 492 widely differing samples.

CryptoDrop, the application they wrote, detected the behavior of all 492 widely differing samples, quickly protecting the majority of the victims' data with as few as zero files encrypted before detection. Not always zero, but in some cases the behavior of the tool was something they were specifically looking for, like the copy and wipe the other thing and so forth. And so they were able to intercept before that happened. So they said: "This result highlights the required actions of ransomware and the effectiveness of our indicators at detecting this type of malware."

In terms of descending attack frequency, PDF files were the most often attacked. Then ODT, DOCX, PPTX [PowerPoint], TXT, then MOV, then ZIP, then MD. And they of course had to look at false positives because if this thing, you know, this is inherently heuristic. It's looking at behavior. And as we know, that can sometimes go off the rails. So they wrote: "While CryptoDrop is effective at quickly detecting ransomware, we note that any evaluation of its real-world utility must also include a discussion on incorrect detection of benign activity. False positive analysis for a system such as CryptoDrop is challenging since its analysis requires changes to be made to a user's protected documents. Techniques used in static malware analysis works, for example, providing a set of known good binaries along with bad ones, but that will not work since CryptoDrop does not analyze binaries for malicious traits, but rather behavior."

Anyway, so they explain their basis for detecting false positives. And so they said: "We evaluated 30 common Windows applications" - and in their document they run through them, and they're things we all know - "on the same virtual machine configuration used to test malware samples and found only one false positive. That false positive was 7-Zip, which was expected, as it reads a large number of files and generates high-entropy output, exactly similar to ransomware."

So these guys have a system which runs in Windows. I saw that somewhere, and I could not find where this thing was available. I'm not sure that it is. But I went from thinking, eh, to thinking, well, okay. This seems like a good thing to have running in the background, just to, like, stop something before it gets loose, especially if they're able to stop it in its tracks as quickly as they allege they're able to based on their analysis.

So I was very impressed. So it is heuristic, looking at behavior. But what ransomware does is enough different from the way we are normally using our computer, with the exception of mass zipping a bunch of files with something like 7-Zip. And if I were running 7-Zip, and this popped up, I'd be glad because it'd be like, oh, cool, this thing's working. Because I know what this is. This is not cryptomalware. And, yes, thank you

very much, I'm going to continue my zip operation. Don't interfere. But, boy, if you weren't doing something like this, and it popped up, it would deserve your attention.

So I'm hopeful that, if not now, in the future, we're going to get something like this in an application because that would be great. And then of course, and they address this on their paper, we're back in the cat-and-mouse game because, if this became widespread and popular, the cryptomalware people would start working on ways around it in the same fashion. But this looks like a strong tool because it's essentially going after the behavior that is exactly what we want to stop, instead of, for example, looking for byte patterns of a virus, and as a consequence having lots of false positives.

And I'll close the podcast by saying that, unfortunately, this Riffle, which I think was the master's thesis of someone at MIT that generated a lot of news a couple weeks ago, looks like a disappointment. We talked at length about how Tor works, the idea being that you choose a path through a number of routers. You get each router's public key. You then encrypt the data you want to send, first for the last router's key. Then you encrypt it for the second to the last. And then you encrypt it for the third of the last, which may be the first.

Anyway, these encryptions form shells of encryption. You then launch it into the Tor network. The first router uses its private key to strip off the outer shell, which exposes the IP of the next router in the hop and so forth, the idea being that no one router is able to see where it's going or get in there. The big problem is, I mean, the fundamental meta problem is the Internet was never designed to provide the kind of anonymity that the Tor network is trying to add after the fact.

So obviously this has been given a lot of attention, and we know that the greatest weakness is a state actor that has visibility into many Tor servers, is able to perform traffic analysis, and essentially catch the traffic going in and associate it with the traffic going out. And as we've talked about this before, what you're best able to do is validate an assumption. So you assume this guy's traffic is coming out over here. That's much easier to verify, that is, to prove the assumption, than to just look at the whole network at once and try to figure out who goes where.

So Riffle said, okay, we're going to do something different. We're going to come up with a system whose traffic cannot be analyzed. The unfortunate consequence of that is it makes it completely impractical. So this has been considered before. This person's master's thesis made it somewhat less horribly inefficient. But essentially, everybody in the network has to always send the same amount of traffic into the network at the same time and receive the same amount of traffic at the same time, specifically to defeat traffic analysis.

And then the stuff you send is mixed in - "shuffled" is the term - with other data, and it moves around the network and comes out. But the idea is that, if all messages are the same length, and everybody is sending the same amount, then you can't do traffic analysis. It also means you have to send a lot more than you want to and receive a lot more than you want to, essentially to cover for everybody else's use of traffic. So, yeah, it would work. But, boy, if people thought Tor was slow, this brings that to a whole new level of pain.

So nothing that we're going to see that we can download and install anytime soon, just an interesting exercise. Maybe something will come from this. This moved the state of the art a little bit further forward than it had been before. But it's got a long way to go before it provides anonymity. And I would argue that, I mean, that's the kind of solution that unfortunately is required, which demonstrates how impractical it is. It's just, wow,

more effort probably than it's actually worth.

**Leo:** And Tor works pretty well. The only issue is of course a government, a state observer with tentacles all over the place might be able to figure out what's going on.

**Steve:** Yeah, yeah.

**Leo:** For the rest of us, I'm not too worried. It's pretty good anonymity. Not perfect. Steve Gibson's at GRC.com. That's where he hangs his hat and where this show originates, from the magic of GRC.com. You'll also find SpinRite, the world's finest hard drive maintenance and recovery utility; all of Steve's stuff, including SQRL, which must be close to coming out; right?

**Steve:** Yeah. If I had more time, I would tell our listeners about some recent improvements. We made a couple of really very cool things. So next week.

**Leo:** Cool. All right.

**Steve:** Because we're out of time.

**Leo:** More than the mic flag? I mean, like substantive improvements. The mic flag's pretty good, though. I'm not knocking the mic flag. You only know that if you see the video. Steve's got his little logo up there. He also, let's see, has audio and human written transcriptions of the show on his website, GRC.com.

I guess we'll a Q&A next week. So if you have a question, GRC.com/feedback, or tweet the man. He accepts the twits, the tweets. @SGgrc is his Twitter handle, @SGgrc. He even accepts DMs because he's a madman. We have audio and video of this show available at our site, TWiT.tv/sn, or subscribe. That way you'll get every episode. Every podcatcher has it. Eleven years and still going strong. Steve, always a pleasure. Have a great week.

**Steve:** Indeed it is. Talk to you next week, my fried. Bye.