
# Twelve and Counting

**Description:** This week we have a Marcus Hutchins update and the back story on the NIST's rewrite of their 15-year-old password guidance. Can DNA be used to hack a computer? Can stop sign graffiti be used to misdirect autonomous vehicles? We discuss the final nail in the WoSign/StartCom coffin, why we need global Internet policy treaties, this week in Researchers Need Protection, a VPN provider who is doing everything right, ElcomSoft's password manager cracker, a bit of errata and miscellany, and some closing-the-loop feedback from this podcast's terrific listeners.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-624.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-624-lq.mp3

---

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We're going to, among other things, of course all the security news, but debunk a couple of stories people saw in mainstream media about hacking a computer with DNA - I fell for that one - and putting stickers on stop signs to confuse autonomous vehicles. It's all coming up next on Security Now!.

**Leo Laporte:** This is Security Now! with Steve Gibson, Episode 624, recorded August 15th, 2017: Twelve and Counting.

It's time for Security Now!, the show where we cover your security and safety and privacy online. And we couldn't do it if we didn't have the best guy in the biz on this subject, Mr. Steven "Tiberius" Gibson. He's sitting right there in the Skype window. Hello, Steve.

**Steve Gibson:** Leo, great to be with you again. We're starting a little late, but it ended up working out well because the leaf blowers made their pass, and the weekly garbage truck has already done its beep beep beep beep.

**Leo:** Oh, we really are late, then.

**Steve:** So it's all worked out well.

**Leo:** Good.

**Steve:** So here we are, Episode No. 624, which we're recording on August 15th. And we recorded Episode 1 on August 18th of 2005.

**Leo:** Oh, boy. Wow.

**Steve:** Which means that between this show and our next show we are lapping ourselves for the 12th time. So I titled today's podcast "Twelve and Counting."

**Leo:** Nice.

**Steve:** Since we are finishing out our 12th year with this 624th podcast. And then on to Lucky 13, which I think will be a busy one. So we've got a lot of fun stuff to talk about and, I think, another great couple hours for our listeners. We've got an update on the Marcus Hutchins versus the government situation. The back story on the NIST's rewrite of their 15-year-old password guidance, which was making news last week; but I thought, you know, we talked about this from the NIST actual document back in June when this happening. But it's generated so much popular attention because, of course, passwords are everything for most people, that I thought, okay, we've got to talk about this a little bit more. And there is sort of a fun back story behind it.

Then the other week's top annoying headline leader was "Can DNA Be Used to Hack a Computer?" So we'll discuss that. And also, can stop sign graffiti be used to misdirect autonomous vehicles? Oh, my lord. We'll talk about that. Then we've got the final nail in the WoSign - the woebegone WoSign/StartCom coffin. Why we need global Internet policy treaties. This week in Researchers Need Protection. A VPN provider who is doing everything right, as opposed to our discussion last week of Hotspot Shield VPN doing everything wrong.

We've got something that worried a lot of our listeners, the ElcomSoft announcement of their password manager cracker that of course targets the top four password managers - 1Password, LastPass, and a couple others. We'll talk about what that actually means. And of course they're the guys that provide all of the rooting stuff and hacking for law enforcement into smartphones and so forth. You know, we've talked about ElcomSoft before. Then we have a bit of errata, some miscellany, and a bunch actually this week, time permitting, of closing-the-loop feedback from this podcast's terrific listeners. So I think two hours from now, as we say goodbye to Year 12, we'll have another great podcast in the can.

**Leo:** Nice. Cannot wait.

**Steve:** So our Picture of the Week, our friend of the show Simon Zerafa sent this to me last week, and I had to preempt it for the cell tower diagram from pre- and post-DEF CON. But this is just sort of - it's fun. It's the security questions fill-out form, and the picture says: "Select three security questions below. These questions will help us verify your identity should you not have access to Google Authenticator app."

So the first question is: "As a child, what did you want to be when you grew up?" And because I'm a developer, and I care about these things, I immediately recognize what's been filled in as the format of a UUID, more easily pronounceable as a globally unique identifier, thus a GUID, because of the grouping of the hex. It's actually 128-bit value which is generated by systems, and there's even an online GUID generator. Some of the groups of digits are 32-bit time, so high resolution. There's a unique system identifier and other stuff. And, for example, things like your Ethernet MAC address, which we know is designed to be globally unique, it often forms part of that.

So the point is that the concept of the GUID or the UUID is that independent people with no communication are able to independently create a token which nobody in the past or the future will almost certainly have, so extremely low collision tokens, and 128 bits gives you a high probability of being able to pull that off. And so anyway, these three questions were not filled out with the actual answers, but with three GUIDs, which are just basically pseudorandom nonsense. And so nobody's going to be able to brute force them. Now...

**Leo:** Where did they get them? You think this is like this guy's MAC address on his, I mean...

**Steve:** Yeah.

**Leo:** There are GUID generators, I guess.

**Steve:** Yes. For example, Microsoft has a GET GUID API, and it just gives you back one. And every time you ask for one, it gives you another one.

**Leo:** Right. I would just use LastPass's password generator and fill it in.

**Steve:** Yeah, that's fine. There are also online generators. And you're right, Leo, just any blob of noise is going to be equally good. You can also go to GRC.com/passwords, as a startling number of people still do. Something took me, by the way, to the Haystacks page the other day, to 4,100 uses of the Password Haystacks page per day. So that's a popular little puppy.

**Leo:** Wow. What does that thing do?

**Steve:** Anyway, so just sort of a fun picture, the idea being, yes, you know, because you and I have been talking about this whole - the problem with this kind of information which has, if you fill out the name of your favorite pet, the street you grew up on and so forth - you can see if you go all the way down to the bottom at the lower right corner, I show the running daily average over the past seven days.

**Leo:** I realize I was just on this page, the Password Haystacks page, and I know why

you're getting a lot of traffic. It's because of this NIST thing. People are going to test "horse stapler," whatever it is, versus - this lets you test how much entropy is in your password, how long it would take to do a brute-force crack of it.

**Steve:** Right. And it sort of gently teaches people about the concepts of the size of your alphabet and so forth.

**Leo:** Yeah, yeah. So it's for fun, but I think that's probably why you're getting traffic right now is because of the thing you're about to talk about.

**Steve:** Very good point. Yeah, that's a very good point. That makes sense. Okay. So Marcus Hutchins, our friend who created the WannaCry sinkhole, who with all good intentions went to Black Hat and DEF CON to just participate, screwed around a little bit in Las Vegas and, as he was at McLaren, which is the Las Vegas airport, got essentially picked up and arrested by agents of the federal government saying "We think you're a bad guy." We've talked about him the last couple weeks. So an update. As we said before, he was going to plead guilty at his arraignment in Milwaukee, and he did. Afterward, one of his attorneys...

**Leo:** He pled guilty?

**Steve:** I'm sorry, not guilty.

**Leo:** You scared me.

**Steve:** He pled not guilty. Yeah, so pled not guilty. Afterwards, one of his attorneys, Marcia Hofmann, called him a hero. Of course she's on his side, so yes. But we all agree. Those people, as we said last week, as we were sharing tweets from people who were shocked and appalled that this had happened to him, who like really know him, that they feel this is ridiculous. And of course the concern is, okay, we don't have all the information that the federal government at this point has. And it's conceivable, I mean, if we want to try to explain, come up with a theory of their case for them, I don't think that's our job, but maybe things in the more distant past, he was different back then. And, you know, I got up to some mischief myself back when I was a teenager.

**Leo:** Well, who didn't, you know, I mean, come on.

**Steve:** That happens. So Marcia called him a hero, said he would be fully vindicated. And in a somewhat interesting change of tone in the government, like their change of tone suggested that something like that may well be the case; whereas at Hutchins's Las Vegas hearing the government used, for example, his appearance at a tourist-focused gun range, as if that had like some nefarious undermeaning, in their attempt to deny him bail. But now the government was amenable to lifting many of the restrictions on his release conditions. He can't go back to the U.K., so I'm sure they have his passport. But he'll be able to live in Los Angeles where his other attorney, Brian Klein, is located. He'll

be able to continue working and can travel throughout the U.S. But as I said, he cannot leave the country to return home.

The only other restriction, which is odd, aside from GPS monitoring, so I guess I don't know if it's an ankle bracelet or something in his pocket, whatever, is that he - and this is the thing that's so weird. The government stipulated he cannot touch the WannaCry sinkhole, which seems oddly random because, like, what? He's going to turn it off and release WannaCry to begin reproducing again? I mean, no. In fact, I mean, I don't even know if he still has control of it. This sort of seems like something where somebody would have said, okay, Marcus, we need to transfer this domain to official government control because this thing is too dastardly to leave in an individual's hands.

Anyway, the government's attorney, whose name I cannot pronounce - Michael, that's easy. Maybe it's Chmelar, looks like C-H-M-E-L-A-R.

**Leo:** All on you today. I'm not going to help you here.

**Steve:** Chmelar described...

**Leo:** Chmelar. Chmelar.

**Steve:** Thank you.

**Leo:** No, I'm making that up. I have no idea.

**Steve:** Every time I pause, you just interject that, Leo.

**Leo:** Chmelar.

**Steve:** Described Hutchins's alleged crimes as "historic," okay, but as in historical, rather than as in historically sized. So that sort of sounds like our theory that we started teasing last week, was that maybe this was something from, like, 12 years ago, or maybe - I don't know how old he is. Maybe five years ago. But like not who he seems to be today. His trial is currently scheduled for this October, but there's some designation that trials can have known as "complex." And the government has made some insinuations that they're going to move to obtain a complex designation for the case which would then likely cause the date to slide, like into next year probably. So we'll sort of keep an eye on this and see where it goes.

**Leo:** Meanwhile, he's in jail. He can't, you know, this is appalling. I'm sorry. Now I'm getting upset.

**Steve:** I don't know when he's going to be released, but...

**Leo:** Is he going to be released?

**Steve:** I think he was released yesterday.

**Leo:** Oh, okay.

**Steve:** This all happened yesterday. And so he can't go home, but he has free range within the U.S. So I'm sure that...

**Leo:** Well, that's what I mean, he's stuck. He can't go back to work.

**Steve:** He can go eat where he wants and not have to have jail food and have scary...

**Leo:** [Crosstalk] chills. Nobody's going to ever come here.

**Steve:** No, no, no, no one is suggesting this is good. And of course you're right. So the first chilling effect was the arrest, and this doesn't make it any better because, I mean, now he's in the system, as they say, and he's got a GPS monitor on his ankle probably, and his life has been wrecked for the foreseeable future. So the good news is this has gotten a lot of attention. And as I said last week, once the facts are known, it's very clear that he will have access to every bit of quality of defense that he deserves. And let's hope he deserves a lot.

**Leo:** Yeah.

**Steve:** So, okay. We talked about when the NIST, the National Institute of Standards and Technology, updated their password guidelines. I jumped on it immediately a couple months ago because the one big issue there was one we've talked about often on this podcast because it was just like fingernails on a chalkboard for me, this ridiculous, apparently arbitrary, you must change your password every N days, or N where N is a small number of months, whatever. I mean, it was just like, why?

I mean, and we've spent time over various points during the last 12 years trying to reverse-engineer this guidance. It's like, what possible value could it have? Yes, long passwords, we understand. Yes, mixed types of passwords, upper/lowercase, special characters, numbers and so forth. In other words, encourage people to try to make better passwords, higher entropy, less easily brute forced. All of that makes sense. But why force people to change their passwords? And even years ago when we first talked about this, there are in Microsoft's code in the group policy, you can do things like you can set policies where passwords expire every X days, and N number of prior passwords are remembered, specifically - and N is like five - specifically to thwart people's ping-ponging between two different passwords every two months or month. It's like, oh, no, we're not going to let you do that, either. We're going to make you have five.

And so people are like, I mean, you know, this is water cooler talk. Once the algorithm

gets out in the coffee room, everyone knows, okay, whatever time of the month it is for your password policy. It's like, I'll change it to add a one, then I'll add a two, then I'll add a three, then I'll add a four, then I'll add a five, and now I'm back to five, and basically I've subverted the entire policy because screw you, IT department. So anyway, the good news is that disappeared after 15 years.

So the funny back story here is where those guidelines came from. They came from a guy named Bill Burr. And now he says, whoops, sorry about those bad password recommendations everyone's been living with for nearly 15 years. Back in 2003, as a midlevel manager at the National Institute of Standards and Technology, Bill was the author of "NIST Special Publication 800-63, Appendix A." Okay, just say "bureaucracy." The eight-page primer advised people to protect their accounts by inventing awkward new words rife with obscure characters, capital letters, and numbers, and to change them regularly.

Today Bill, who is now 72 years old and retired, said that: "Much of what I did I now regret." The Wall Street Journal wrote: "The document became" - the original document from 2003, wrote The Wall Street Journal - "became a sort of Hammurabi Code of passwords, the go-to guide for federal agencies, universities, and large corporations looking for a set of password-setting rules to follow." The problem is that the advice ended up being largely incorrect, Mr. Burr said. Change your password every 90 days? He laments what we have often noted on this podcast: Most people being periodically forced to make an unwanted and unneeded change will make only minor changes that are easy to guess.

So a couple months ago in June, as we covered at the time, Special Publication 800-63 got a thorough rewrite, jettisoning the worst of these password commandments. Someone who's there, Paul Grassi, an NIST advisor who led a two-year-long do-over, said the group thought at the outset, two years ago, when this project began, that the document would require only a light edit. But two years later, Mr. Grassi says: "We ended up starting from scratch."

The new guidelines, which are already filtering out, thank goodness, through to the wider world - and two months ago, at the time when this happened, I said, okay, yay. Now if this is like, for some reason, the tablets from on high, then fine, at least they've been edited. And now we have new tablets; and people can say, well, this is what the NIST guidelines are, so don't fire me. Thank you. So they're already filtering out through the wider world, writes the Wall Street Journal, and drop the password expiration advice and the requirement for special characters, interestingly.

Mr. Grassi said: "Those rules did little for actual security and had a negative impact on usability." Of course many people, they put in their normal password - unfortunately they have one, "monkey" - and then the system says, oh, you must have a digit and a special character. So they add an exclamation point and pound sign on the end, just to satisfy the annoying incoming password filter. So, okay, that's gone.

So a little on Mr. Burr's background. He once programmed Army mainframe computers during the Vietnam War. He had wanted to base his advice at the time on real-world password data, a sound approach. We've looked, for example, at disclosed password lists, and from those breaches we've learned a lot about how horrible passwords are and the surprising dropping in the popularity of the word "monkey" as everyone - I think it was used, you know, it was way up there for a while. Not quite up at where "password" is, but close.

But back in 2003, when Burr was tasked with this job, there wasn't much available real-

world password data to work from. So he was under pressure also to publish this guidance quickly because of course we have to have it now and then be stuck with it for 14 years, right or wrong. So he asked the computer administrators at the NIST if they would let him have a look at the actual passwords on their network. Well, they refused to share them. And he said, citing privacy concerns: "They were appalled that I even asked."

So with no empirical data on password security to be found, he leaned heavily on a whitepaper written in the mid-'80s which was pretty much the only source material that he could find. A guy named Cormac Herley, who's a principal researcher at Microsoft who the Wall Street Journal pulled in for some comment, said that, collectively, humans spend the equivalent of more than 1,300 years each day typing passwords. Well, we don't because we have wizards that do that for us. "Microsoft once followed the Burr code for passwords," said Cormac, "but no more. The NIST rules were supposed to give us randomness. Instead, they spawned a generation of widely used and goofy-looking passwords such as Pa$$w0rd or Monkey1!." Cormac said: "It's not really random if you and 10,000 other people are all using it."

So anyway, I got a kick out of some of the coverage. The idea that these rules were changing, of course, generated lots of interest in the wider public press. And it was nice to see, okay, how did we get stuck with this bad advice for so long? Well, now we know.

And speaking of bad, boy, I tell you, this was - the headlines were crazy for this one, and I have three of them here: "You can hijack a gene sequencer by hiding malware in a DNA sample." Or "Biohackers Encoded Malware in a Strand of DNA." Or "Scientists Hack a Computer Using DNA."

And then that particular one starts out: "In what appears to be the first successful hack of a software program using DNA, researchers say malware they incorporated into a genetic molecule allowed them to take control of a computer used to analyze it. This biological malware" - oh, help me - "was created by scientists at the University of Washington in Seattle, who called it the first 'DNA-based exploit of a computer system.'" Oh, wow.

Okay. To carry out the hack, researchers led by a team "encoded malicious software in a short stretch of DNA they purchased online. They then used it to gain 'full control,'" in quotes - and believe me, there's a lot of detail to be discussed here in a second - "over a computer that tried to process the genetic data after it was read by a DNA sequencing machine."

Okay. So last Thursday all of this ridiculous, over-the-top nonsense was the popular press's interpretation of a 15-page paper which appeared at the USENIX security conference last Thursday. The paper was titled, and I have it, and I read it, "Computer Security, Privacy, and DNA Sequencing: Compromising Computers With Synthesized DNA, Privacy Leaks, and More." Okay, so that's not a modest title. Here's what these guys - okay, well, first I'll explain what they did. But here's the abstract from the top of this 15-page research paper.

**Leo:** And in my defense, because I did this story last week, and a lot of people did, all the coverage preceded Thursday's paper.

**Steve:** Right, right.

**Leo:** So you have an advantage that we didn't have. All we had was the abstract. Go ahead.

**Steve:** Okay.

**Leo:** And I fell for it. I did.

**Steve:** I missed your coverage. So the abstract says: "The rapid improvement in DNA sequencing has sparked a big data revolution in genomic sciences, which has in turn led to a proliferation of bioinformatics tools. To date, these tools have encountered little adversarial pressure." Okay. I'm completely - so I'm not saying this is not useful, but this is not what the press said. "This paper," they say, "evaluates the robustness of such tools, if or when adversarial attacks manifest. We demonstrate for the first time the synthesis of DNA which, when sequenced and processed, gives an attacker arbitrary remote code execution." Oh, boy. Yes, but with more caveats than even I could cover on the podcast. But we'll get there in a second.

"To study the feasibility of creating and synthesizing a DNA-based exploit, we performed our attack on a modified" - that's an important word - "modified downstream sequencing utility with a deliberately introduced vulnerability." Okay, so they created something that would be deliberately vulnerable which, when it encountered their purpose-built DNA snippet, would crash. Oh, heavens. Okay, well.

"After sequencing, we observed information leakage in our data due to sample bleeding." Which is really not the term you want to use when you're sequencing DNA, but anyway. "While this phenomenon is known to the sequencing community, we provided the first discussion of how this leakage channel could be used adversarially to inject data or reveal sensitive information." Which is a little bit of a stretch, but okay.

"We then evaluate the general security hygiene of common DNA processing programs" - certainly that's useful - "and, unfortunately, find concrete evidence of poor security practices used throughout the field." Bravo. So if nothing else, this will drive the implementers of software to be a little more concerned the way the implementers of servers need to be for the robustness of their solutions. And they conclude the abstract: "Informed by our experiments and results, we develop a broad framework and guidelines to safeguard security and privacy in DNA synthesis, sequencing, and processing." Noble goals.

Okay. So what's going on here? There's a utility on SourceForge that's open source called "fqzcomp," also known as the "FASTQ" compression utility. It was designed to compress DNA sequences. So these guys downloaded the source and deliberately broke the code by creating a short static buffer which was deliberately too small to contain their 177 base pair DNA so that, when that program attempted to compress the DNA they provided to it, that synthetically and purposefully shortened buffer would overflow. And their modified fqzcomp version used a simple two-bit DNA encoding scheme. As anyone who's looked into genetics knows, there are four nucleotides that have just been given the letters A, C, G, and T. So four is a happy number. So the four nucleotides were encoded A as 00, C as 01, G as 10, and T as 11. Thus this allows packing pairs of bits into bytes so that you can get four nucleotides encoded, each nucleotide bringing two bits with it into an eight-bit byte.

Okay. So, yes. Their two-bit scheme, so to speak, coupled with their deliberately hacked reduced buffer size, did indeed allow them to place their own binary bits into non-buffered space, that is, into the overflow region past the buffer. However, even given all of this cheating manipulation, their exploit only functioned 37.4% of the time since only an error-free gene sequencing would produce a running exploit. Oh, darn, you know, your shell code has to not have bugs in it, literally, or bad genomes. And only 76.2% of the DNA sequencing of this synthetically short little snippet was error free. So the sequencing process isn't sufficiently robust today to even allow you to reliably transcribe code from DNA into bits, that is, just, what, a little over two thirds of the time, 76.2%.

But then there's the problem that DNA strands have two ends. And the sequencing might begin from either end. So of course this cut the 76.2% success in half again, down to 37.4 because, if you ran the wrong end in, you ran the back end in first, you'd get your shell code backwards. Now, maybe if you were really a good hacker you could make the shell code work, regardless of which way it was encoded, which would be a cool hack. But theirs didn't.

So what we really have here is a somewhat interesting, contrived, synthetic demonstration of how an inherently very low reliability attack could be created using DNA as the code-carrying agent, but almost certainly only if the DNA sequence processing pipeline had serious flaws that would cause it to crash quickly when processing non-contrived DNA. In other words, it's difficult to see how any such flaws, such as what they put into the code to make this happen, could survive in the real world without immediately crashing upon encountering actual DNA, which of course would drive the developers of the code to fix it.

So anyway, that's the story behind, oh, my god, DNA can crash computers. Yeah, except it can't actually. And you know, Leo, it occurred to me that this whole merging of medicine, DNA, and computers is kind of a bit surreal. And one of the things we've seen in the past is how various disciplines, when they intersect, kind of cross-pollinate with their terminology. And one wonders in this case how much terminology from these disciplines might get combined because, for example, you could imagine that incontinence might someday be considered a buffer overflow.

**Leo:** Oh, please. I mean, I think the story stands as a cautionary tale that data files - and this is how I used it on the radio show - can, if there's an error in the rendering program, be used to infect...

**Steve:** In the interpreter.

**Leo:** In the interpreter. What they didn't, or I didn't see in the mainstream stories, or they didn't reveal until they gave their paper, was, A, that they hadn't modified DNA; and, B, more importantly, they modified the interpreter to make it flawed so it didn't - I mean, that's a silly proof of concept, but it is a proof of concept. But it's not proving anything we don't already know, which is that, if you have a bad metafile interpreter, a JPEG can be used to infect a computer. If you have a bad PDF reader, a PDF can be used. We've seen these attacks happen. And so it isn't that farfetched to say, well, I guess if you're running sequencing software on your PC, which they do, you could theoretically somehow malform a DNA strand to pwn the computer.

**Steve:** Correct. And again...

**Leo:** They didn't do it.

**Steve:** As I said at the top, I think it's cautionary and useful. But again...

**Leo:** They didn't actually do it, yeah.

**Steve:** Right, exactly. So we have another - the week's top clickbait headlines, which were used to mislead and worry people. And I put two headlines at the top of the story here: "A self-driving car can be easily hacked by just putting stickers on road signs." And the subhead of that one was "A team of experts showed that a simple sticker attached on a sign board can confuse any self-driving car and potentially lead to an accident." Except that's not at all what the team of experts showed. I mean, it's like there's not a shred of that is accurate.

And the other headline that I liked was "Researchers hack a self-driving car by putting stickers on street signs." Except, no, that's not what they did, either. Not even remotely close in this case. The DNA story was more accurate than this one.

So in fairness, the audience for these articles is not our typical Security Now! listener. But they do cause concern and put unwarranted doubt out into the ether, which is bad. So in this case this team of expert researchers from four U.S. universities - the U. of Washington, U. of Michigan at Ann Arbor, Stony Brook University, and the University of California at Berkeley - got together and, okay, so now listen to the actual title of their paper, which was "Robust Physical-World Attacks on Machine Learning Models." Okay? Doesn't mention wheels or cars. Doesn't say attacks on cars, or even one car, or even a shopping cart. It says "attacks on machine learning models."

And yes, this has some relevance to the real world because we believe that autonomous self-driving automobiles also use machine learning models. So cars have, to some degree, that in common with what these researchers did. And they were interested in researching the question of spoofing cars, but that isn't what they actually did.

Okay. So my point is at this point it's very different from what we've discussed, for example, in the past of, like, for example, commandeering that Jeep's management network and forcing its driver off the road and into a ditch. I mean, that actually happened. This hasn't. This didn't.

Okay. So the abstract of their paper reads: "Deep neural network-based classifiers are known to be vulnerable to adversarial examples" - and I should just stop and say, notice now in both of these stories the notion of adversarial attack is presented. And this is one of the best changes that we're seeing happening. I mean, it's surprising that it takes as long as it does. But now there's becoming this mature understanding that just getting the code to work is different from having it always do what you want it to, very different. And so you have to challenge the code.

So anyway, so they say: "Deep neural network-based classifiers are known to be vulnerable to adversarial examples that can fool them into misclassifying their input through the addition of small-magnitude perturbations." Which is "I have a degree" speak for, like, tiny blemishes in the image can have outsize effects. "However," they write, "recent studies have demonstrated that such adversarial examples are not very effective in the physical world. They either completely fail to cause misclassification, or

only work in restricted cases where a relatively complex image is perturbed and printed on paper." In other words, real cars, you know, people have tried to spoof real cars. And while we've talked about some of the limited successes they've had, there is some robustness there.

So they say: "In this paper we propose a new attack algorithm" - which they named "Robust Physical Perturbations," and so it's RPP - "that generates perturbations by taking images under different conditions into account." So they're going to try to strengthen the attack concept. "Our algorithm," they said, "can create spatially-constrained perturbations" - meaning, again, small - "that mimic vandalism or art" - a.k.a. graffiti - "to reduce the likelihood of detection by a casual observer.

"We show that adversarial examples generated by RPP achieve high success rates under various conditions for real road sign recognition by using an evaluation methodology that captures physical world conditions. We physically realized and evaluated two attacks, one that causes a stop sign to be misclassified as a speed limit sign in 100% of the testing conditions" - but hold on because we need to discuss what the testing conditions are because they weren't real - "and one that causes a right turn sign to be misclassified as either a stop or added lane sign in 100% of the testing conditions."

Okay. So because this news generated these hysterical headlines, these guys had to do a FAQ, Frequently Asked Questions page in order to back people away from the hysteria that this was creating. So in their FAQ they ask themselves, "Do you attack a real self-driving car?" Answer: "No." "Okay, what did you attack?" And they answered: "We attacked a deep neural network-based classifier for U.S. road signs. A classifier is a neural network, in the context of our work, that interprets road signs. A car would typically use a camera to take pictures of road signs, and then feed them into a road sign classifier." In other words, something that reads road signs.

They write: "To the best of our knowledge, there is currently no publicly available classifier for U.S. road signs." In other words, there are autonomous driving vehicles and experiments, but those are corporate enterprises that have proprietary algorithms that they value highly, and they're keeping to themselves. So they couldn't use those. "Therefore," they write, "we first built our own neural net consisting of three convolutional layers followed by a fully connected layer. We then trained our network on the LISA dataset, a U.S. sign dataset comprised of different road signs like stop, speed limit, yield, right turn, left turn, et cetera. Our final trained road sign classifier accuracy was 91% on the test dataset." Okay. So it got it right a little, just slightly over nine out of every 10 times when re-shown signs that it had been trained on. So sort of a synthetic result. But again, it was correct for their purpose.

So then the next question: "What are your findings?" They said: "We show that it is possible to construct physical modifications to road signs, in ways that cause that trained classifier to misinterpret the meaning of the signs. For example, we were able to trick the classifier into interpreting a stop sign as a speed limit 45 sign, and a right turn sign as either a stop or added lane sign. Our physical modifications for a real stop sign," they say, "are a set of black-and-white stickers." And in the press coverage there were lots of pictures of this, basically stop signs with some black-and-white rectangles sort of stuck at seemingly arbitrary positions, but not arbitrary.

So then they finally say, just sort of for real-world-ness: "What resources does an attacker need?" Now, first of all, this would be an attacker attacking their neural network, not any known actual autonomous vehicles. So they say: "An attacker needs a color printer for sticker attacks; a poster printer for poster printing attacks. The attacker would also need a camera to take an image of the sign he wishes to attack."

So then: "Based on this work, are current self-driving cars at risk?" "No. We did not attack a real self-driving car. However, our work does serve to highlight potential issues that future self-driving car algorithms might have to address." No argument there. Finally: "Should I stop using the autonomous features - parking, freeway driving, et cetera - of my car? Or is there any immediate concern?" "We again stress," they say, "that our attack was crafted for the trained neural network discussed above. As it stands today, this attack would most likely not work as-is on existing self-driving cars."

Okay. So once again, they created a relatively simple neural network, that is, a four-layer network which they trained, and they were then able to understand and monitor. So then they gave it a stop sign and moved black-and-white rectangles around the stop sign while this thing kept trying to classify the stop sign. And as soon as the arbitrary shape and size and position rectangles caused a failure of this four-layer simple neural network, they said, "Ah-ha, we just fooled it." Okay, yeah.

So anyway, this is another definitely useful research. We definitely hope that all manufacturers of actual road sign-classifying AI will see this research. I'm sure everybody who should see it, it has come to their attention, and they're probably busily explaining why their neural network AI classifiers would not fall to this kind of attack. So, interesting and useful; but again, the headlines were crazed on this, saying, yes, you put a couple stickers on a stop sign and Teslas will drive right through and not stop at the stop sign and kill their owners. No. Sorry.

And the researchers realized, whoops, we weren't clear enough, so they added this FAQ page to say, okay, no, we didn't attack a car. We didn't attack a real network. We don't have any. No one will give theirs to us. So we made one that wasn't very good to start with, which we were then able to scrutinize while we did things to it until we made it guess wrong. So, I mean, and nine out of 10 times it guessed wrong anyway, even when given a good sign that it had learned. So anyway, yes. Cars at this point are not in danger from graffiti on road signs. But useful.

The final nail in the WoSign/StartCom coffin: Microsoft has joined Mozilla, Google, and Apple in the abandonment of trust in WoSign and its StartCom subsidiary. So it's finally game over for those clowns. We've talked about, back years ago, when it first came to light the really surprising low quality of their automated frontend for their certificate minting. So this and the Symantec debacle should serve as a useful and cautionary tale for all other presently widely trusted certificate authorities, which is what you do really matters. That is, not what you say. Not all the certificates and assurances you provide. What you do is what matters because that's entirely why they're in the business of signing certificates, and the only way they're in that business is if they are trusted. So if what they do breaks trust, they're out of business.

So in Microsoft's TechNet blog of August 8th, Microsoft writes: "Microsoft has concluded that the" - I mean, and they're the last to join the party, so yay - "that the Chinese Certificate Authorities WoSign and StartCom" - which as we know is a subsidiary of WoSign - "have failed to maintain the standards required by our [Microsoft's] Trusted Root Program. Observed unacceptable security practices include back-dating SHA-1 certificates, misissuances of certificates, accidental certificate revocation, duplicate certificate serial numbers, and multiple CAB Forum Baseline Requirements violations.

"Thus, Microsoft will begin the natural deprecation" - that word struck me as odd. Natural deprecation? Okay - "of WoSign and StartCom certificates by setting a 'NotBefore' date of 26 September 2017." That happens to be Tuesday exactly six weeks from today. "This means," they write, "all existing certificates will continue to function until they self-expire. Windows 10" - and I'll come back to this in a second because the whole posting

only addresses Windows 10 - they write, "will not

trust any new certificates from these CAs after September 2017," so from October on. "Microsoft," they write, "values the global Certificate Authority community and only makes these decisions after careful consideration as to what is best for the security of our users." Okay.

But what was not addressed in that posting was the status of the earlier, still supported, and still in the majority despite Microsoft's every effort, Windows 7 and 8.1 operating systems. Note that this sort of conditional certificate acceptance tweak, unless the facility was presciently already built in, requires custom modification of the system's certificate interpreter. That is, by default, the certificate start and end dates are checked. So you've got to do something other than that if you want to do something like conditionally allowing certificates that were only made before the end of September.

So users of earlier Windows operating systems cannot simply, if we're abandoned by Microsoft, cannot simply remove those root certs from their own trusted root stores in their Windows OSes, since StartCom's certificates were once quite popular and are still being sold today - today and for the next six weeks, doubtless - with two or three years in the future expirations. So it's not the certificate owner's fault that the company from which they were obtained screwed up, although I guess I would hold people somewhat responsible for having purchased certificates from StartCom anywhere in the recent past after it was known that they were such poor managers of trust.

Okay. So what this means, though, is that any StartCom certificate issued within the next six weeks will still be honored by Windows, even Windows 10, for the next two or three years. Okay. So to be responsible to its users, Microsoft really must similarly protect users of its earlier and still supported operating systems. Failing that, we would be forced to defensively abandon use of IE in favor of Chrome or Firefox on those earlier platforms because those browsers are remaining proactive, even on earlier versions of Windows.

And out of curiosity I went over the StartCom this morning. And it's business as usual over there. You wouldn't know, looking around, that they're not long for the world. But in fact the boom is rapidly being lowered on them because, unless they were to start backdating their newer certificates' start date - and remember this wouldn't be the first time they had done that. But boy, this would end it. That would immediately extinguish what little remaining shred of forbearance the industry has for them. But from this date in six weeks they will be unable to ever again sell any certificates which would be honored by Windows 10 or these other browsers, starting six weeks from today.

So the problem is that, because StartCom certificates will still be valid for as many as two or three years, other versions of Windows need to be informed of this same thing. I don't know that Microsoft isn't going to do this. But Microsoft wants to pretend that nobody else is running anything other than Windows 10, so they just don't talk about 7 and 8.1. Hopefully, we will get an update. Maybe it's already there; it has been delivered. Or it'll come next month or sometime. But anyway, I hope they...

**Leo:** It doesn't come in the browser. It's a Windows Update file.

**Steve:** Correct. Correct, because IE and Edge both use the underlying Windows crypto systems. Chrome has historically. I think they still do. But the browser is able to look at the certificate, even though the OS processes it. So Chrome, in looking at the certificate,

gets to say, whoops, that's a StartCom cert. And the NotBefore date, meaning the issuance date, we've decided we're not going to trust. So the browser is able to conditionally trust a certificate that the underlying server still fully trusts.

And so that means that what Microsoft will do with Windows 10 is they're going to push that conditional trust down into the kernel, down into the underlying crypto system, which is why it's not just - they can't just change a certificate. They have to arrange, you know, they have to change the code to educate it about the new policy for StartCom and WoSign certs. Whereas Firefox carries their own completely internal certificate management system. Chrome uses the underlying OS's, but does inspect the cert that the site has given it. And so it's able to say, you know, to essentially do a blacklist of certs that would otherwise be trusted.

Okay. So some news out of the U.K. sort of put me in mind of why we need, why we're going to need, why the only solution to the problems that are beginning to surface and metastasize are going to be some sort of global Internet treaties. BleepingComputer reports last Monday that British lawmakers - or reports that: "Last Monday, British lawmakers filed a statement of intent regarding" - and again, this is just a statement of intent, so not legislation yet, so it has an unclear future as anything not yet signed into law does - "regarding proposals for improvements to what they call the Data Protection Act, the DPA, with a focus, interestingly, on criminalizing anonymous data reidentification, imposing large fines for cyber incidents and more user protections for British online netizens," writes BleepingComputer. "The modifications are part of the U.K.'s effort to comply with the EU's General Data Protection Regulation, the GDPR, that's set to come into effect in May of 2018."

And that thing has some real teeth. There's been some discussion of it, I think, in the Security Now! Forum in the GRC newsgroups. And I mean, I have to do some things with GRC's ecommerce system in order to make sure I'm in compliance with the GDPR. I don't remember now whether we've talked about it on this podcast. But, I mean, it doesn't matter where you are. If you have customers in the EU, the EU is asserting that they have teeth to go after anybody anywhere who is not complying. So it's certainly a chill.

**Leo:** We have a fan who has written a book on - I wish I had it with me, I think it's at home - on GDPR compliance. It's a little book to get you started. We'll send you a copy.

**Steve:** And believe me, you and I both need one because...

**Leo:** Well, do I need one? I don't have customers. I don't collect information.

**Steve:** This thing is so horrifying when you read it, it's like, wait a minute.

**Leo:** We have server logs, so I guess that counts.

**Steve:** Believe me, Leo, if you have a server online, I mean, unless you're Wikipedia, I think you're probably - you need to just make sure. And I don't know if little fish will matter. But, boy, there's no doubt that Google and Facebook and the big players are looking at this, thinking, okay, we've got to decide what we want to do about this.

Okay. So the U.K.'s data protection act has this new statement of intent. One of the high points is this notion of anonymous data reidentification. Okay. But the bill would add many provisions to the GDPRs. For example, "Make it simple to withdraw consent for the use of personal data." That is, require people who have personal data to make it simple for their users to withdraw consent for the use of their personal data. "Make it easier and free for individuals to require an organization to disclose the personal data it holds on them." So all good for visibility.

"Allow people to ask for their personal data held by companies to be erased. Require explicit consent to be necessary for processing sensitive personal data. Enable parents and guardians to give consent for their child's data to be used. Expand the definition of 'personal data' to include IP addresses, Internet cookies, and DNA." And, finally, "Make it easier for customers to move data between service providers." So this would have a chilling effect on much of the way business is conducted today.

And unfortunately, as is too often the case when aggressive new legislation meets aggressive new technology, problems in practice arise. For one thing, it's easy to drop the gavel on a law that's not possible or practical to implement. We've been discussing this; we've been discussing sort of around this issue with a whole question of law enforcement somehow having access to encrypted information, especially if that information is encrypted with perfect forward secrecy so that the keys are constantly changing, and there's no history of them. It's like, so the legislation can say whatever it wants, but the technology doesn't make that possible.

Another perfect example would be the requirement, which we've discussed, and there has been proposed legislation, and it's somewhere in the pipeline, that ISPs retain a log of everything each of their subscribers individually does and everything they do out on the Internet. Okay. Well, it's easy to write the law. It's easy to say that. But as we know, it's next to impossible for any ISP to actually pull that off, even if they wanted to, which they most assuredly don't. They just, as we know, want to charge for the transitive subscriber traffic across their proprietary networks, thank you very much. They don't want that responsibility.

But there are governments that say, well, sorry, but we need the information that you're in a unique position to get, not quite understanding that they're actually not. That information is not gettable. So similarly, in this instance, it's a noble ideal to imagine criminalizing the reversal of deliberately anonymized identity information, but how exactly is that reduced to practice? It is, after all, the entire business model of several of the world's largest Internet entities. It underlies the somewhat awkward Hobbesian bargain we have made with these entities in exchange for accepting their free offerings. They track and explicitly deanonymize, even in the face of many of their users' explicit request that this not be done. So good luck with telling Google and Facebook and now Microsoft as a Service that this is conduct unbecoming and could become criminal.

So paraphrasing from BleepingComputer's report, they say on top of the GDPR provisions the data protection bill includes an extra provision, the creation of a new criminal offense for when someone intentionally or recklessly reidentifies individuals from anonymized or pseudonymized data; and, in practical terms, answering the question who belongs to this cookie because all browser cookies are inherently pseudonymous, and you're not supposed to know who that is. But that's what Google and Facebook are. So the DPB reads: "Offenders who knowingly handle or process such data will also be guilty of an offense. The maximum penalty would be an unlimited fine."

In BleepingComputer's coverage, they quoted a Dr. Lukasz Olejnik, and he is an independent cybersecurity and privacy researcher who is an affiliatee of Princeton's

Center for Information Technology Policy, who on one hand applauds the U.K.'s efforts, writing in his blog last Monday, quote: "The U.K.'s GDPR implementation may have visionary traits in that it goes beyond merely implementing the GDPR as just a legislation. The U.K. will introduce new criminal offenses, among them reidentification." But then he adds, oh, by the way: "There are several issues with the banning of reidentification. First, it won't work. Second, it will decrease security and privacy.'"

The biggest problem, in Olejnik's view, is that there's no effective way to enforce it in practice. But wait. Since this is Google's and Facebook's entire business model, enforcement seems pretty simple. After the legislation is enacted, if it should ever actually see the light of day as law, the U.K. can simply give Zuck a call and say, you know that legislation that became law yesterday? How are you guys still in business? So secondly, though, adds Olejnik, the new legislation would stifle security and privacy research, which often and must often deliberately reidentify anonymized data in day-to-day research. The DPB statement did mention protections for journalists and whistleblowers, but did not provide any details.

So anyway, this brings us all the way back around to the need for transnational treaties. And this is the tension that we've been watching grow over the last couple years. We have a global network with valuable and desired content being provided free of explicit charge to us, in exchange for a sacrifice of absolute, that is, of our absolute privacy and anonymity. We can feel comfortable, and many people do, that Google is going to be responsible, and Facebook will be responsible. But our privacy is not absolute in return for their services for which they don't charge us.

And of course these are massive global Internet-centric enterprises operating across national boundaries. If regulation is to be imposed, that regulation cannot practically be disparate in every region of the globe. You can't have a global network with entities like Google and Facebook spanning the globe which also spans governments, each with their own quirky laws that these entities operating within those countries all have to individually abide by.

So we're still in the early days. But I think we're beginning to struggle with the big questions of encryption, privacy, and anonymity for all users of this incredible global network of ours. It's clearly an important and necessary conversation for us to have. But the very globalness of the Internet which creates so much of its value requires unified global regulation. And I think a treaty mechanism is probably the way we do that. I don't, you know, I'm not a legal scholar, so I don't know the details of that. But presumably someone somewhere is thinking about this. We can hope.

**Leo:** So I just have a question on the NIST story. I know we're going back in time.

**Steve:** Yeah, yeah.

**Leo:** Our advice still stands. And I guess that's what changed. I was surprised that they didn't say use the special characters because I always use.

**Steve:** I always do.

**Leo:** What I do is I tell LastPass, use every character you can - special characters, numbers, upper, lower, whatever. The key is long, right, and totally random.

**Steve:** Well, yes. But the difference is these guidelines are still assuming user-remembered passwords.

**Leo:** Oh, I get it.

**Steve:** And that's the real problem is that now we know you have to have a separate password for every site, or you risk cross-compromising non-compromised sites. So all of our listeners, who are all password manager fanatics, none of this, you know, bears...

**Leo:** None of it applies; right.

**Steve:** None of it applies because we're using...

**Leo:** Because we don't have to remember it.

**Steve:** Exactly. We're using high pseudorandom content gibberish every site we visit.

**Leo:** Right, even for secret questions.

**Steve:** Yes.

**Leo:** But there is one password you have to remember, and that is your password vault password.

**Steve:** Good point. And actually we're going to come to the story in a minute here about the brute forcing of the password vaults.

So this week in Researchers Need Protection, Techdirt wrote up a very nice piece that features one of our frequently mentioned security people, Troy Hunt, titled: "Company Storing Families' Personal Data Blocks Users/Researchers Informing It of a Security Flaw." Techdirt wrote: "It must be repeated over and over" - and everyone knows it's my refrain - "people who discover security flaws and report them are not the enemy. And yet," writes Techdirt, "company after company after company treat security researchers and concerned users like criminals, threatening them with lawsuits and arrests rather than thanking them for bringing the issue to their attention." And of course we've got the Hungarian ticketing, public transportation ticketing system from a couple weeks ago, too, as another example.

This company, Kids Pass, a U.K. company providing discounts for families attending restaurants, theaters, and amusement parks, had a problem. And, boy, any user could

access any other user's personal information just by altering numbers appearing in the userIDs in the URL. Oh, I mean, I think it's like there's a top 10 worst practices. This was number four, the idea of sensitive information in the URL. No. In other words, another glaring example of atrocious web application design which itself should be outlawed.

So a concerned user told security researcher Troy Hunt about the flaw. And so Troy tweeted: "Just this weekend I had a Twitter follower reach out via DM" - I'm sorry, Troy posted in his blog. "Just this weekend I had a Twitter follower reach out via DM, looking for advice on how to proceed with a risk he'd discovered when signing up to Kids Pass in the U.K., a service designed to give families discounts in various locations across the country. What he'd found was the simplest of issues, and one which is very well known: insecure direct object references [is the technical term for this]. In fact, that link shows it's number four in the top 10 web application security risks, and it's so high because it's easy to detect and easy to exploit.

"How easy?" Troy writes, "Well, you can count; right? Good, you can hack. Because that's all it amounts to, simply changing a short number in the URL," which of course points you to a different user. And it's like, oh, now I can see their data. "Troy told the user to stop doing anything, including accessing other users' information" - good advice, you don't want to actually be breaking the law just unintentionally - "and to immediately inform the company. The user did as instructed, contacting the company via Twitter direct message." Okay, so a private communication, Twitter direct message.

"Shortly thereafter, the user informed Troy that Kids Pass had blocked him on Twitter. Troy then made an attempt to speak to someone at Kids Pass, only to find he'd been blocked, as well, most likely for having the gall to retweet the concerned user's message about the security flaw. The responsible, ethical approach, notifying a company of a security flaw as soon as possible, was being treated like some kind of trollish attack on Kids Pass's Twitter account. From all appearances, the company simply wanted everyone to shut up about the flaw, rather than address the concerns raised by a user.

"It was only after Troy asked his followers [he has many] to contact the company on his behalf [thus flooding them] that Kids Pass finally unblocked him and told everyone the 'IT department' was looking into it." So yay for Troy. "However, that belated reaction doesn't make up for the initial reaction, of course; and Kids Pass has shown it has little interest in addressing security flaws until the problem becomes too public for it to ignore. Troy points to a blog post by another security researcher who informed Kids Pass last December about its insecure system, including the fact it sent forgotten passwords in plaintext via email to users." So no technology on their backend. This developer heard nothing back, finally publishing his discoveries in July, so last month, giving them ample time to fix it. They didn't.

"If you want people," writes Techdirt, "to be good web citizens and report breaches and flaws, you can't treat them like irritants or criminals when they do. Securing users' personal info is extremely important, but some companies seem to feel they should be able to handle it however they want and mute/sue/arrest those who point out how badly flawed their systems are." And our listeners will remember that, when one was pointed out at GRC, someone found a fault in my form processing that technically allowed cross-site scripting. It turns out that there were other limitations. But I thanked the person, I told everybody on the podcast, and I fixed it. So that's the way you do it.

As I've always said, people are responsible for their policies. They're only responsible for the way they deal with their mistakes. But mistakes happen. So a company like this, wow. I mean, yeah, we have no regulations that make it illegal to conduct yourself this poorly. And there's no robust reputation system in place. If there were, it wouldn't

matter. People would still go, oh, hey, look, it's free, and dump all their information in with no idea that that URL is their account number, and that anybody can put that in and retrieve all their info.

So, I mean, again, very much like I was talking about last week with the Chinese webcam that mapped itself through UPnP and established open waiting service ports on the public Internet, I mean, like the worst possible way to do this. And I gave an example of a perfect, simple, correct way to do it. So there are already well-known good ways to solve all these problems. But not only do companies not do it, but when someone says, hey, you know, this is bad over here, they get attacked by way of thanks so often.

So on the heels of last week's horrific Hotspot Shield VPN story, where our listeners will remember it turns out that the free-to-download clients are doing all kinds of noxious privacy-violating things as researchers had found, we have the flipside that I wanted to share, both because I've had a lot of our users asking me about this particular provider, but more so because their posting demonstrates all of the right stuff. And our listeners will recognize and appreciate the honesty and integrity it displays. So I'm going to share the entire thing - it's not overly long - because this is important, not only for current and prospective users of TunnelBear, but because there are also lessons here for any and all other high-integrity VPN providers.

I've got a link to TunnelBear's blog posting in the notes, as well as a link to the third-party audit which they have now done several times. The headline on their posting is "TunnelBear Completes Industry-First Consumer VPN Public Security Audit." And they write: "Consumers and experts alike have good reason to question the security claims of the VPN industry. Over the last few years, many less reputable VPN companies have abused users' trust by selling their bandwidth, their browsing data, offering poor security, or even embedding malware.

"Being within the industry, it's been hard to watch. We knew TunnelBear was doing the right things. We were diligent about security. We deeply respected our users' privacy. While we can't restore trust in the industry, we realized we could go further in demonstrating to our customers" - and, I would argue, to the rest of the industry - "why they can, and should, have trust in TunnelBear." Or for the rest of the industry, what real proper conduct looks like. "Today," they write, "we'd like to announce TunnelBear has completed the consumer VPN industry's first third-party public security audit. Our auditor, Cure53, has published their findings on their website, and we're content with the results." And they provided a link.

So a bit of history: "In late 2016," they write, "we hired Cure53, a respected security company, to do a complete audit of our servers, apps, and infrastructure. Using a white box approach" - as opposed to a black box approach, where what's inside is a mystery. So "Using a white box approach, they were given full access to our systems and code. Our original plan was to use their findings internally to confirm we were delivering on our promise to secure your browsing and proactively identify vulnerabilities. However, the recent crisis of trust in the VPN industry showed us we needed to break the silence and share Cure53's findings publicly. Today we're sharing a complete public audit which contains both the results from last year and the results from the current audit.

"As the auditor, Cure53's opinions and findings are their own, with the results being published on their website." So clearly TunnelBear is trying to establish a bit of an arm's-length relationship here to assure us of the veracity of these findings. "TunnelBear," they write, "was given the opportunity to provide feedback on the report before it was published, where we felt findings were inaccurate or irreproducible. As is the case of

most security audits, Cure53 was paid for their work. We wouldn't expect any cybersecurity company to spend a few hundred hours auditing our code for free.

"What were the results? If you've already looked at the results, you've seen that the 2016 audit found vulnerabilities in the Chrome extension that we weren't proud of. It would have been nice to be stronger out of the gate, but this also reinforced our understanding of the value of having regular independent testing. We want to proactively find vulnerabilities before they can be exploited. We hadn't intended to publish the 2016 results. However, we're hoping the security community has appreciation for our candid transparency in the 2016 report and for demonstrating our investment in security over time. All findings discovered in the 2016 audit were promptly addressed by TunnelBear's engineering team and verified to be fixed by Cure53.

"In the June 2017" - that is, the audit that they're writing about that just happened last month - "we were more content with the results. All vulnerabilities represented low-risk findings. As Cure53 put it: 'The results of the second audit clearly underline that TunnelBear deserves recognition for implementing a better level of security for both the servers and infrastructure as well as the clients and browser extensions for various platforms.' All findings discovered in the 2017 audit have also been addressed by TunnelBear's engineering team, with only informational findings remaining." They say you can read the full report on Cure53's website.

And, finally, "Our ongoing commitment to security," they write, "our plan is to earn trust and move the VPN industry in a new direction around transparency. While many VPN companies will continue to live in obscurity with claims for protecting your security, it's our hope that by completing the industry's first third-party public security audit, experts and consumers alike can be sure that TunnelBear delivers on its security promises.

"If we've learned anything from this audit, it's that good security needs constant reevaluation." Amen. "Annual public audits will become routine to help us quickly identify vulnerabilities and demonstrate transparency in an industry where trust is sorely lacking. In the coming months we'll share more announcements, industry insights, and how-tos to give you the information you need to make the right choices about your security." And they sign off saying, "Grizzly Regards, The TunnelBear Team."

And again, props and bravo. I mean, that's - I just like this on the heels of Hotspot Shield VPN as the way it's done right. A company that already had the right stuff and contracted with an independent third party to check their stuff, and problems were found. They addressed them immediately; and then they thought, you know, six months later, let's check again. And so fewer problems, I mean, nothing significant was found. And they said okay, let's explain who we are.

And I like their notion of the fact that most VPN providers, many are unclear about their commitment to user privacy. They just say, oh, you know, "protects your privacy," but nothing to back it up. Here we have somebody that was allowed in, with no axe to grind, cross to bear, axe to grind, saying okay, this looks good. And so bravo. As I said, a lot of users have asked me about TunnelBear. I've not looked any further. I don't know about their plans or their pricing or anything else. But this, as an ethic for a VPN service provider, I don't know how you do any better than this. I know proXPN was a longstanding VPN sponsor.

**Leo:** So was TunnelBear, by the way. TunnelBear's a sponsor, as well.

**Steve:** Oh, they are now?

**Leo:** No, they have been.

**Steve:** Oh, okay. I didn't know that. So, nice.

**Leo:** Yeah. Once a sponsor, always a sponsor.

**Steve:** Okay, good.

**Leo:** Disclaimer-wise, anyway.

**Steve:** Right. So I've not looked at their plans, but I wouldn't have a second thought about either of those guys because we know that their hearts are in the right place. So bravo to them.

Now, many users - I think this is the last one of our news topics. Yes. Many users worried about, once again, some other headline news. This was ElcomSoft's blog posting, which began: "One Password to Rule Them All: Breaking into 1Password, KeePass, LastPass, and Dashlane," arguably the four leaders in the password manager territory. I have a link to their full posting in the notes. And I've just snipped a tiny few pieces out of it.

They wrote: "We've just updated ElcomSoft Distributed Password Recovery" - which is a product of theirs - "with the ability to break master passwords protecting encrypted vaults of the four most popular password keepers: 1Password, KeePass, LastPass, and Dashlane. In this article we'll talk about security of today's password managers and provide insight on what exactly we did and how to break into encrypted vaults."

Okay. So skipping all of that, this amounts to an offline GPU-accelerated, brute-force password recovery. So essentially they've done enough reverse engineering of those four password managers to figure out what the algorithms are which map the user-supplied password to the key, which is then used to decrypt the user's password vault. And they then turned that into the highest performance code they could. So, for example, they write: "Different password managers employ different approaches to security. As an example, LastPass generates the encryption key by hashing the username and master password with 5,000 rounds of PBKDF2 using SHA-256."

So we've talked about PBKDF2 in the past, and the SHA-256 hash. So the idea being that it is an iterative process which turns the users' password into pseudorandom gibberish, with the requirement that you put the same password in every time, you get the same gibberish out every time. And it also, because it's done 5,000 times, it slows it down way more than if it was just done once, if it was just hash the password through SHA-256. Instead, it's ground on a lot longer.

And they say: "...while 1Password employs even more rounds of hashing." Although interestingly, 1Password was more attackable than LastPass. We'll get to that in a second. They write: "This is designed to slow down..."

**Leo:** Slow it down, buddy, slow it down.

**Steve:** I need some more tea. Maybe I do need some caffeine.

**Leo:** Slow down.

**Steve:** "...to slow down brute-force attacks, and it almost works," they write. "Granted, these are still nearly an order of magnitude less secure, say, than Microsoft Office 2016 documents; but even this level of security is much better than nothing." They write: "Therefore, this is the benchmark. We've added RAR5 and Office 2016 to the chart for comparison. Higher numbers represent higher recovery speeds." And Leo, I've got their chart in the show notes on the page that this leads into. And it's a little interesting.

So, for example, this is using an NVIDIA GTX 1080 GPU to drive the hashing engine. And with the default settings, LastPass passwords can be brute forced at 44,800 per second. 1Password can be brute forced at more than twice the speed, at 95,000. So certainly we believe what they wrote about the higher number of iterations, but it still is faster. So maybe 1Password is not using PBKDF2, it's just iterating the hash, which makes it extremely acceleratable by a GPU.

**Leo:** So the bigger number is less secure.

**Steve:** Correct. Correct.

**Leo:** It's odd that Office 2016 is so secure.

**Steve:** I know. Isn't that interesting?

**Leo:** It surprises me, yeah.

**Steve:** Yeah. Yeah, so that's guesses per second. And so the least secure was Dashlane at 129,000 guesses per second. But, okay, now, before anyone gets hyperventilated here, remember that, if you have a sufficiently long and random, really high-entropy password, well, then you can figure out how many bits it has. For example, you could use the Password Haystacks to give you the size of the search space, and then divide that by your password manager's guesses-per-second figure, assuming this acceleration. And that will tell you how long it would take to search the entire password space, cut that in half for the average search length.

And now, for a bit of context, so for example LastPass, 44,800 guesses per second. Office 16 the strongest of everything, 7,300 guesses per second. So for some context, because this sort of local brute-force attack is always a concern, the technology I designed for SQRL's PBKDF, that is, SQRL's password-based key derivation function, is deliberately highly acceleration resistant using the memory hard Scrypt function, and dynamically self-adjusting using what we call the "dynamically iterative EnScrypt function." It yields a

maximum brute-force rate of 0.00333 guesses per second. Which is to say, one guess every five seconds. And it is very difficult to make it go any faster. So it is possible to do even better, if you really want to. And for SQRL, of course, I did want to.

So I guess the takeaway is they didn't crack an of these products' code. But they have added to their line a super, as much as they could, a turnkey GPU acceleration-enhanced brute-forcing capability. So this is the kind of thing, if the FBI were to grab somebody's phone or laptop or something in a warranted search, and the user refused to divulge the password, they would, probably do have a contract with ElcomSoft, and they would update to the latest version and put it to work cracking that person's master password, password manager database. And depending upon the quality of their password, and only the quality of their password, they would get those results.

Oh, and note that was 44,800 guesses per second at the LastPass setting of 5,000. We've talked about that setting. You can turn it up higher, if you want, and see if it noticeably slows things down. SQRL, in return for making brute forcing a ridiculous challenge of futility, does take, the first time you use it, five seconds. I mean, you sit there. There's a little progress bar that runs across only once, the first time you use it, when your screen is unblanked, or you've switched users, or you haven't used it for a configurable length of time. And you can turn those settings on and off. So you pay the price once to prove to SQRL that you're you. And then afterwards we allow you to use what we call the "quick pass," which is just a short prefix of that password and only takes a second, so just, you know, no time at all.

So it's a workable compromise that doesn't sacrifice security anywhere. So it can be done. And if you were worried about LastPass, or I don't know if the other password managers allow you to crank up their settings, but you might explore turning it up and seeing if, on a given platform and browser, it becomes burdensome. Probably doesn't. And so it might be worth going to a higher setting and waiting a few seconds for it to say, yeah, you guessed right. Or you didn't guess, you knew your password.

I wanted to follow up a little bit with a little bit of errata from last week. Many of our listeners confirmed that the send.firefox.com site that we talked about is working under Safari, including the listener who originally reported that it wasn't. So either something weird happened at his end transiently, or maybe, because it is a web-based service, they may have fixed it between the time he tried it and the time I talked about it. So anyway, I did want to follow up with everyone. And in fact I think you, Leo, tried it while we were on the air last week, and it seemed to work. So a lot of other people said, yeah, it works for me.

Last week I talked about the mystery, the ongoing mystery of my $2.04 Amazon S3 service billing. And I said I was going to try to get some statistics. Unfortunately, I only put the query in when I was beginning to assemble the podcast the previous evening, so it wasn't - and they kept saying, "You're going to need 24 hours." And, boy, they sure didn't give it to me any earlier than that. But as a consequence, I have it now. So here's what I know. I am currently storing 88GB at Amazon, for which I am paying $2.02 per month at 2.2 cents per GB per month, which is the price for the first 50TB of storage. I'm not bothering with Glacier, which I understand is one fifth of the cost. But that would almost be sad.

**Leo:** They would charge you 2 cents.

**Steve:** There was no charge for data transfer because I'm just, as I said, mostly I'm just

sending the podcasts up there and other - I have a bunch of encrypted and compressed directories that I also send there, and other stuff. But, for example, in one month there were - I was charged 1 cent for 560 PUT requests and 1 cent, that is to say, one penny, which is worth now more than a penny, for 852 GET requests. So I'm not a heavy user, 88GB, so I'm not storing images of multi-terabyte systems or anything. But I'm storing the data that I care about for $2, and not even using Glacier.

So anyway, I did want to close the loop on that because I'm quite happy. I mean, Amazon has the, what is it, it's a ridiculous number of nines, 99 point - it's much more than five nines. It's, I don't know, like nine nines, 99.99999999% uptime. And so I know there are alternatives. There are cheaper solutions. But you know me, I'm at Level 3 because I like their bandwidth. And Amazon's got a good service.

So I got a nice tweet, I wanted to mention, from Richard Philips. He is the author of the Rho Agenda trilogies. And he just said: "Hi, Steve. Thanks for the Rho Agenda mention on last week's show. I had a lot of fun with the final novel in that series, 'The Meridian Ascent.'" And I wrote back and thanked him for his note and told him that, as soon as I get to it, I will read it and probably mention it again because I really do enjoy his books.

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

**Steve:** Huh?

**Leo:** That's nice.

**Steve:** Yeah. And "Colossus: The Forbin Project," a number of our listeners found it. It exists over on Archive.org.

**Leo:** Really.

**Steve:** Yes.

**Leo:** Wow.

**Steve:** And there is a downloadable MP4, which is, boy, 409MB, I think. And so, yup, you found it, Leo. And I got the...

**Leo:** Oh, it's high quality. It looks good.

**Steve:** So what I wanted to tell our listeners is to please, it is browser-based. So you can just fire it up in your browser, if you don't want to download the MP4, but you can. Just watch the first five minutes. It's just fun. It's Dr. Forbin walking solo through the halls of this underground massive computer facility as he brings it to life. And I don't want to say any more because it's just, I mean...

**Leo:** This looks like an analog circuit. I hate to say it.

**Steve:** Oh, I know. Well, so its release date was April 7, 1970.

**Leo:** Oh, there's paper tape.

**Steve:** So I was a freshman in high school when this came out. And so it's really all...

**Leo:** It's fun.

**Steve:** It's all the fault of the movie, yeah. Anyway, just the first five minutes. I looked at the clock after Dr. Forbin had walked out of the facility, and other things had happened. And I thought, okay, I just need to tell our listeners, if they're curious, you don't have to download it, just look at it in your browser, five minutes.

[Clip in background]

**Leo:** I love the computer sounds, too. Isn't that great?

**Steve:** I mean, this is classic old B-movie.

**Leo:** That's so funny. Wow.

**Steve:** So I did have a note, actually from somebody who's been working with me on SQRL over in the SQRL newsgroup. He was having some weird behavior that he suspected might be a consequence of his system. So he posted in the newsgroup. First he said: "There are a couple of unrelated things that have mysteriously stopped working over the last two or three days." He said: "I know my aging hard drive is slowing down. So I'm going to shut down and SpinRite the hard drive."

And then he followed up in that thread to his own comment. He wrote briefly: "Ran SR Level 2 and then Level 4. No sector or other issues flagged. System boots and runs noticeably faster and smoother now. Apps launch noticeably faster now. Per past experience, this effect will last one to two weeks. Then things will slow down again." He said: "I am looking for a replacement hard drive."

And so thank you, Dan. I appreciate your sharing that. And I should say that SQRL is now - the thing I've been working on for the last several months, the install, update, and remove facility, is essentially nailed. We found one problem yesterday with a user, a tester who has the unbound local DNS resolver running in his machine. And there is a problem with that, but we figured out what it is.

Anyway, essentially we've already moved on and are in the process now of cleaning up the few remaining issues. I mean, it's done. It's functional. The spec is finished. I mean, like the spec's definition is now finished. So my plan is to spend X amount of days in the

future, I have a to-do list that I've been making all along the way of things I wanted to remind myself to get back to and check. I will do that. And then we'll be talking about it on the podcast, and it will be released for everyone to play with.

And then while that's happening I will get back to work on the web side. I don't want to delay allowing people to understand what it is that we've created here. But I need to get it documented. Apparently there is an Android client going that I wasn't aware of. Jeff in the U.K. has got his iOS client. I'll have one. And it is running under Linux, also. So we have those bases covered. But I wanted to mention that, as is the case when odd things are happening, and this was what Dan was addressing, was if we can't trust brand new software, the question is, okay, is there something wrong with the software? Or is there something wrong with the system on which it's running? And he began to suspect that, well, other things were not working right, and his machine was seeming suspiciously slow. So it might not be a problem with this new code that we were testing, but rather the platform it was being tested on.

I wanted to mention, though, I mean, the real takeaway here is this demonstrates what many SpinRite users have found through the years, which is it will keep a drive alive. It will bring them back. But ultimately this is hardware which is dying. And again, it's like you give yourself an IV - oh, gosh. What was that from "Silicon Valley," young blood, you know, my blood boy. They were getting transfusions from young teenagers and feeling all energized. Anyway, so SpinRite is your blood boy for your PC. The point being, though, at some time in this battle between SpinRite refusing to allow your drive to die prematurely, it will ultimately lose. Just as I'm currently in perfect health at 62, well, yes. But this is a battle I'm ultimately going to lose at some point. I hope not soon.

But the point is one of the things you need to be aware of is that SpinRite lets you get every last ounce of use out of your system. But when the drive finally refuses to come back to life, well, it has the final say. So take heed of these sorts of things. Take heed of things seem funny. Now they're okay after running SpinRite. Oh, look, they got funny again. Oh, I'll run SpinRite again. Okay, yes, but your blood boy's getting tired. So don't push it too far.

We've got a bunch of closing-the-loop stuff. I'm going to save most of them for next week. But there was one that I just, oh, my goodness. Actually two. A C. M. Au Yong tweeted: "HTTP Error Code 418." Now, for those who aren't as versed in HTTP error codes as some of us, everyone knows the 404. That's the famous "this page doesn't exist" error code. And there are other ones. There's a 300 series for redirects like moved permanently, moved temporarily, and so forth. But so in the HTTP spec there are all these defined error codes. And from time to time, engineers being what we are, a little humor gets injected. There actually is an Error Code 418. And the format of these error codes is like HTTP - so it'll say HTTP 404 Not Found. So the 404 gives you a numeric item to lock onto, and then Not Found is like the English version of that because you can imagine maybe some website or some browser would just print on the page the English tail at the end of that error. Okay, well, Error Code 418's English part is "I'm a Teapot."

**Leo:** Short and stout. Wow.

**Steve:** So if you make a query to a web server, and it gives you back HTTP Error 418 I'm a Teapot, well, you know that the web server has a sense of humor. Anyway, so this tweet was the result of a Reddit thread because apparently this is about to be removed from Node.

**Leo:** Awww.

**Steve:** And a person said, "We've got to do something."

**Leo:** No, don't take that out.

**Steve:** We don't want to lose our teapot.

**Leo:** There may actually be history. I mean, remember there was the Cambridge Coffee Pot was one of the first spy cams, many, many years ago. It may actually be that there was a teapot server. And maybe that was the - don't ask me anything. I'm a Teapot. I can't - send me no GET commands.

**Steve:** That'd be perfect. I'm a Teapot. So Evron Baldwin retweeted something from I Am Devloper, and that's his handle. And this is just so good. This has got to be, well, certainly the Tweet of the Week. So what I Am Devloper tweeted was a pro tip which is brilliant: "Set your password to be the word 'incorrect,' and every password message becomes a password hint." Your password is incorrect. Oh, you're right. Thank you for the reminder.

**Leo:** That's like the Hobbit password. What was it? Say friend and pass or something like that? I can't remember what it was. It's one of those hide in plain sight passwords.

**Steve:** Yup.

**Leo:** Yup. Love it. Steve, we've come to the end of another fabulous edition of Security Now!.

**Steve:** At the end of another fabulous year.

**Leo:** Ah. Year 13 begins next week.

**Steve:** Next week.

**Leo:** Tuesday, 1:30 Pacific, 4:30 Eastern, 20:30 UTC, if you want to watch live. Many of you do. And if you do, you should probably be in the chatroom at irc.twit.tv so you can join the kids in the back of the class. They knew about I'm a Teapot, I think. Not required, of course, because we have many ways to watch the show on demand at your leisure, including, of course, Steve's site, GRC.com. He has audio and transcriptions of each and every show at GRC.com. While you're there, pick up a

copy of SpinRite, world's best hard drive maintenance and recovery utility. You never know when it will come in handy.

**Steve:** It'll keep your drive alive until it runs out.

**Leo:** Keep your drive alive. You'll also find information about SQRL, Password Haystacks, ShieldsUP!, all sorts of freebies. Steve gives everything away except for that one bread-and-butter program, SpinRite.

We have our copies of the show at TWiT.tv/sn, audio and video, should you choose to watch Steve gesticulate. And my suggestion, my strong suggestion is subscribe to the show because a lot of times we mention previous episodes. You might want to go back in time. And the best way to do it is to have a complete catalog of all 600, what is it?

**Steve:** 624 and counting, baby.

**Leo:** 624 episodes. All 12 years. The complete set. Steve, thank you so much for a great 12 years. Here's to 12 more. Well, I guess we'll get to 10 more, maybe.

**Steve:** Okay, my friend. Thanks so much.

**Leo:** Take care.

**Steve:** Talk to you next week in the beginning of Year 13 with Episode 625.