The Equifax Fiasco

Description: This week we discuss last Friday's passing of our dear friend and colleague Jerry Pournelle; when AI is turned to evil purpose; whether and when Google's Chrome browser will warn of man-in-the-middle attacks; why Google is apparently attempting to patent pieces of a compression technology they did not invent; another horrifying router vulnerability disclosure including 10 zero-day vulnerabilities; an update on the sunsetting of Symantec's CA business unit; another worrying failure at Comodo; a few quick bits; an update on my one commercial product SpinRite; answering a closing-the-loop question from a listener; and a look at the Equifax fiasco.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-628.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-628-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. And he, you know, it's funny when something like the Equifax breach happens, I don't know about you, but I just wait all week just to hear what Steve has to say about it. Well, your wait is over. Equifax and a lot more coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 628, recorded Tuesday, September 12th, 2017: Equifax Fiasco.

It's time for Security Now!, the show where we cover security, now, with this guy right here, Steven Gibson of the GRC, Gibson Research Corporation, GRC.com. Steve is a security researcher, discoverer of spyware. He also wrote the first antispyware tool, and he's been doing this show now for 13 years. We've been doing this show for 13 years, since we were young men, all about security and privacy. Hi, Steve.

Steve Gibson: Yes, Leo, it was about 12.5 years ago we were in Toronto between filming breaks, and you said, "What do you think about doing a weekly, maybe like a half an hour podcast on, like, the week's security events?" And I said, "A what cast?" No one had ever heard of that before.

Leo: You hadn't even heard of podcasts at that time, yeah, wow.

Steve: Yup.

Leo: I see you're growing the moustache back.

Steve: Well, I don't know what I'm doing. I'm lost.

Leo: Well, Jerry Pournelle, I never saw Jerry Pournelle without a moustache, and I was even looking at old video of him from the '70s, and he had that pencil moustache that he was famous for.

Steve: Yup. And his bolo tie, remember, I don't know what you call it, sort of he had like the big...

Leo: Yeah, bolo tie, that's it, yeah.

Steve: Yeah, exactly. Anyway. So today's podcast, 628, is of course titled "The Equifax Fiasco." And so we'll pull all this together. A couple of our foreign listeners said, "What's a credit bureau?" So I guess they do things differently elsewhere. So but this is of course the big security news of the week, so we're going to cover that. And we wanted to talk briefly about Jerry Pournelle, our very good friend and colleague who passed away Friday, peacefully in his sleep, at the age of 84, about a month after celebrating his 84th birthday.

Also about when AI is turned to evil purpose, which unfortunately apparently is going to be a thing before long. Whether and when the Google Chrome browser will warn of man-in-the-middle attacks. There was a lot of interest among our listeners about this news that Chrome is going to do it. It turns out that's a little bit of a red herring. And it kind of has to be because this is being done so much now. Also why it appears that Google is attempting to patent pieces of a compression technology that they did not invent, which I imagine you and Jeff and Stacey will expand on tomorrow. But there's sort of some interesting technical angles to it.

Another horrifying consumer router vulnerability disclosure, this time including 10 zero-day vulnerabilities. An update on this forthcoming sunsetting of Symantec's certificate authority business unit. Another worrying failure of operations over at Comodo. We just keep coming back to those guys. A couple quick bits. And then we're going to - I've got to close the loop on one listener question, and then we're going to talk about the Equifax fiasco in detail. So I think another - oh, and the perfect Picture of the Week that ties in, but we'll get to that in a second.

Leo: Nice. I can't wait. Always enjoy those. For those of you listening live, yeah, we know Steve is having a little glitching still. But for those of you on the podcasts, apologies again for last week, but I think we'll have a good-sounding podcast for you because Steve's recording his end locally. He'll probably sound better than ever before. We'll see.

Steve: I don't think it'll help, but okay.

Leo: Your moustache will be crystal clear 4K UHD. All right, Steve. Let's see that image you were talking about.

Steve: Oh, boy. The Picture of the Week instills real confidence in Equifax.

Leo: Oh, boy.

Steve: On their security encryption page, it starts: "In the United States you can order all Equifax products online with confidence using Netscape and Internet Explorer."

Leo: What? Netscape?

Steve: Oh, yes.

Leo: The Netscape?

Steve: Netscape Navigator, Leo.

Leo: What?

Steve: Under SSL and 128-bit Encryption: "If you have Netscape Navigator, simply select 'Help' from the menu bar." And down lower it explains about the broken key that Netscape Navigator shows if you don't have encryption.

Leo: Is it possible there are still people using Netscape? No.

Steve: This company is on the leading edge of security technology as...

Leo: Holy cow. I can't - that's unbelievable.

Steve: Yeah. Talk about being inattentive. So, yikes.

Leo: This is on their current website?

Steve: Yeah.

Leo: Oh, lord.

Steve: Yeah.

Leo: Well, there were lots of mistakes that they made in the various pages they put out.

Steve: Oh. Oh, boy.

Leo: I'm really looking forward to hearing your dissection of this one.

Steve: Yeah, yeah, yeah. So I did just want to take a moment for a little bit of memoriam to our very good friend and colleague Jerry Pournelle. I knew him pretty well, back in the early days of the PC industry. And I mentioned before that his wife Roberta is a real hugger. And it always used to seem to really annoy him that whenever the three of us would run into each other at a computer show, Roberta and I would give each other an extra-long hug. And he'd sort of stand there looking a little flustered, kind of like, okay, when is this going to be over?

Leo: I love it.

Steve: So he was born on August 7, 1933, so he had his birthday, his 84th birthday last month. And his son wrote last Friday, he said: "I'm afraid that Jerry passed away. We had a great time at DragonCon. He did not suffer." So, and of course, as we know because he was around TWiT during the time that he was fighting that brain cancer that he did bounce back from and recover from, I mean, so the guy was tough. And, boy, what a pivotal personality through the whole formation of the early PC industry and Byte magazine, Chaos Manor, and all of his contributions over the years.

Leo: Yeah, he was a great, treasured contributor on TWiT - 20 different TWiTs. We also did two Triangulations with him, which if you want to know more about his life and times, they were great. He had some great things to say.

Steve: Yeah. And he and Larry Niven, another sci-fi author, put themselves together, and they wrote a number of books together. My all-time favorite, though, was "The Mote in God's Eye."

Leo: Yeah.

Steve: In fact, I think all of their books, come to think of it - and it may have been Larry's influence, but I know that Jerry was the same way - there is like a core concept that is, you know, sitting around coffee or something, they come up with an idea that's really good. And it just - it stands alone. And then they encapsulate this perfect, pure surprise in a fully fleshed out sci-fi setting.

And that's what "The Mote in God's Eye" is. You're reading along, and it's really interesting, and they're leaving clues along. So it's sort of a mystery, kind of. But there's,

like, a big "oh, my goodness" that, like, jumps out at you about three quarters of the way through. So just a really pleasant read. And for what it's worth, if anyone has never had the chance to read "The Mote in God's Eye," it is a sci-fi classic and definitely worthwhile. So Jerry, you will be missed.

Gizmodo covered an interesting story, although I did the math that they did and came to a different conclusion than they. Although it shouldn't surprise us that the bad guys will be weaponizing AI for their own nefarious purposes. Last year, Gizmodo writes, two scientists from the security firm ZeroFOX conducted an experiment to see who was better at getting - well, "who" in quotes - better at getting Twitter users to click on malicious links, humans or AI.

"The researchers taught an AI to study the behavior of social network users and then design and implement its own phishing bait. In the tests, the artificial hacker was substantially better" - and I'll take issue with that in a second - "than its human competitors, composing and distributing more phishing tweets than humans, and with a substantially better conversion rate."

The AI, which was named SNAP_R, "sent simulated spear-phishing tweets to over 800 users at a rate of 6.75 tweets per minute, luring 275 of those 800 into the phishing scheme. By contrast, Forbes staff writer Thomas Fox-Brewster" - we've often quoted him on the podcast - "who participated in the experiment was only able to pump out 1.075 tweets a minute, as opposed to 6.75, making just 129 attempts and luring just 49 users." So as I'm reading this, immediately my rough math says, okay, wait a minute, that's about the same conversion rate. Different gross rate, but the same fraction.

Anyway, Gizmodo finishes, saying: "Thankfully, this was just an experiment, but the exercise showed that attackers are already in a position to use AI for their nefarious ends. And in fact they're probably already using it, though it's hard to prove. In July [a couple months ago] at Black Hat USA 2017, hundreds of leading cybersecurity experts gathered [as we know] in Las Vegas," Gizmodo writes, "to discuss this issue and other looming threats posed by emerging technologies. In a Cylance poll held during the confab, attendees were asked if criminal hackers will use AI for offensive purposes in the coming year, to which 62% answered in the affirmative."

Okay. So I guess this should not be surprising. But I checked the math, and the "capture rate" or the "conversion rate" was almost the same for both the human, that is, Thomas Fox-Brewster, and the AI. It was 34.4% versus 38%, and those are not statistically significant differences for such a small sample size. So, I mean, so that difference with such a small sample set could have just been chance, luck of the draw.

So the takeaway here is that it's certainly worrisome that an AI could be at least as good as a human because that's zero effort, and it's automatable, and it's infinitely scalable. So not good news for us. But again, also not very surprising. But we're sort of just putting on our radar because we do put things on our radar, and they come back to bite us almost invariably.

In fact, we talked about this whole credit freeze thing two and a half years ago, Leo. And we'll get to that later. But I thought, okay, yeah. Our listeners who took our advice then weren't in any danger from this, or at least comparatively so. There's even a bit.ly link I created that I verified was still valid: bit.ly/freezecredit. And it still works. And that's two and a half years old.

Leo: Does it link to all three of the reporting agencies?

Steve: It takes us to a page which has been maintained which walks a consumer through the processes for all three bureaus, yes. And so it was good then, and it's still good today. So again, this is one of the fun things we get to do on a 12-plus-year podcast is see how these things, these threats and issues evolve over time.

Another sort of a red herring, as I said at the top of the show, was this news that Chrome, Google's browser, of course, would soon be warning us of software that performs man-in-the-middle attacks. Well, first of all, I didn't know how it could do that because, as we have now been discussing a lot, many enterprises are doing so-called "middleboxes." That is, they're doing HTTPS proxying on their border in order to be able to inspect the traffic that's moving into their Intranet and protect themselves from it in case it's malicious because, as we know, post-Snowden, the world is going encryption.

So in digging into the story, which confused me, it turns out that there was a summer intern, just this past summer, who for her project looked at adding this technology to Chrome, which will get incorporated. But it seems like what is actually going on is that there's a gray area in improperly or poor-written middleboxes which can do some sort of breakage. And I wasn't able to get any details. Maybe when it actually rolls out we'll be able to come back to this and understand what's going on. But I did want to note to our listeners because I got a bunch of people saying, hey, look, this is going to prevent snooping. It's like, no, it won't.

And remember, the other really unanswered question is whether, like what Chrome does because remember that Chrome pins the signatures of all of Google's certificates in itself. And this is how so many fraudulently issued Google certs have immediately come to Google's attention because any person who uses Chrome, who goes to a Google asset which is coming from a fraudulent certificate, immediately sends a beacon back to Google saying, hey, I just saw a Google cert that you guys didn't produce.

So the point is that middleboxes are synthesizing their own certificates on the fly. Thus the signatures would not match, and Chrome would not be happy. So it may be that those middleboxes whitelist the Google domains and pass them through without trying to intercept them in order to keep Chrome from - I just don't know. But that would be one way to do it.

So this feature that Chrome is adding is some sort of quality metric where - and again, I'm just - unfortunately there isn't anything that I was able to find that had the nitty-gritty, which everyone knows that I prefer to work from. But it's if some threshold of errors are occurring, then the user is warned with a big intercept screen, big scary thing that comes up and says something to the effect of your firewall or AV is mis-installed, is what it says, and inducing errors, you know, you need to fix that.

So the idea is it will be tolerant of the, like, well-written, properly performing interception, but not poorly done interception. And again, I hate that I don't know exactly what that means. But hopefully at some point we will get some clarification. It says this will appear in Chrome 63, and so on the horizon, and as a result of some work by a summer intern. But again, I did want to back our listeners away because many people picked up on this and said, hey, that looks great. It's like, well, okay. Apparently it's just to help clean up the industry. Which is, again, a good thing. As we know, Google often uses its clout in order to discipline people who are behaving badly.

And this story is interesting, too. Google has been accused of trying to patent public domain technology. This surrounds an arithmetic data compression approach which was deliberately placed in the public domain by its developer, a Polish assistant professor who goes by Jarek Duda, D-U-D-A, who has accused Google of trying to patent technology he invented and deliberately placed into the public domain, specifically so that no one could own it. I mean, his research was paid for by his university, and he just wanted to give this away and keep it from being locked down.

We were talking just recently, when we were talking about the decompression of the firmware in the Intel Management Engine and Huffman coding, how it uses variable-length tokens to represent variable frequency occurrences. That's a form of arithmetic encoding and arithmetic compression. This guy has a much, much more advanced form of, as it's called, "entropy coding," which is known as an - it's an Asymmetric Numerical System, or ANS, which he's been working on from 2006 through 2013. And a number of companies have jumped on this and adopted it. There's a variation called tANS and another one called rANS. It's used in Apple's LZFSE compressor. Facebook uses this in their Zstandard compressor. And Google is employing it in their Draco 3D compressor.

So patents are weird. And from this distance it's impossible to judge. I don't know what Google is thinking, or if this professor is feeling very proprietary about his technology, and in fact Google may have had some innovations they've added. There is a third party involved, which is there's an international patent group that is siding with the professor, believing that what Google's claims are alleged to be are just already in the public domain, well known, already in practice and so forth. But again, as I said, patents are tricky. Sometimes you initially go in with a deliberately overbroad application; and then, as we were talking just last week, sometimes you just get lucky - well, unfortunately for us - and the Patent Office says, yeah, okay, fine. And then of course you have to litigate that in court, which would not be easy to do when Google is on the other side of that.

Or, more often, the U.S. Patent and Trademark Office will come back and say, well, claims 13, 14, and 15 we will allow; but one through 12, no, we're going to disallow those. And so there's some, you know, it's why it takes many years, typically, and there's some back and forth. So it could be that Google has innovated on top of the existing technology, which many patents do grant. We'll have to see how this thing plays out. Apparently Google is being silent.

There's some belief in the industry that they have pulled this back, and they're going to rework it in order to make it more clear what they've done and what they haven't. So we'll see how this evolves over time. It's not absolutely clear where things stand although, as I said, some independent judges have looked at this and don't feel that there's anything that really merits the label of invention and intellectual property protection.

Add to our list of routers behaving badly a recent state-of-the-art D-Link router, the 850L, which is a Wireless AC1200 Dual-Band Gigabit "Cloud" Router. I put "cloud" in quotes because it has built-in cloud features. It's got something called "mydlink Cloud Services" which allow you to access, view, and control the devices on your home network from anywhere. So that's a powerful capability, and you would like the router to be secure.

Unfortunately, a researcher initially examined this router for his participation in a contest and was quickly appalled by what he found. Then he found 10 different worrisome zero-day vulnerabilities of varying concern, which I'll run through here in a second. He tried to responsibly disclose to D-Link, who never worked with him, kept silent. He waited for months with no apparent action. And then these was some disclosure where they didn't

acknowledge that this came from him, but said, well, a researcher happened to discover a couple problems. And so it's still not clear. And in the most recent updates they still haven't fixed everything. So anyway, the thousand-foot view of this is, if you have a D-Link 850L, I read several accounts saying just unplug it because, well, you can see why here.

So there's two different firmware releases, a version A and a version B. A is not protected in any way, so firmware images can be forged by attackers. There's no protection. B added password protection. Unfortunately, the password is hard-encoded and never changed, so that doesn't really afford any protection. Okay. So in the revA firmware, both on the WAN and the LAN side - and this researcher in enumerating these problems was careful to differentiate the two because, of course, as we know, LAN-side vulnerabilities are not good. If something gets into your network, or depending upon like how large your network is, if you are an enterprise using one of these 850L D-Link routers on your perimeter, then any employee in a sufficiently large corporation would have access to the router from the LAN side.

Of course we know the WAN side is a much bigger concern because that's the world that has access. And as we've seen, and we've been talking lately or recently, there are search engines like Shodan that can find all these things. So the revA firmware exposes both on the public WAN and the private LAN cross-side scripting vulnerabilities with PHP files, which can be exploited as a consequence of these cross-side scripting vulnerabilities to steal authentication cookies from the router. That's in the revA firmware. On the revB firmware, again both WAN and LAN, it's possible to retrieve the admin password for the router to gain then full access using this custom mydlink cloud protocol.

And apparently without breaking the D-Link's terms of use, this researcher found vulnerabilities which could allow attackers to abuse this mydlink cloud protocol and register the router to their own accounts, to then gain full and unfettered access to the user's D-Link cloud and all the internals of their network. So just abominable security. It turns out also that this cloud protocol is fundamentally weak, so independent of firmware version there's a WAN side exposure.

The researcher wrote that the mydlink cloud protocol is little more than a basically TCP relay system with no encryption by default. Traffic is sent over TCP to the Amazon servers that host the cloud without encryption. So it's fully susceptible to sniffing and to man in the middle because, remember, if you don't have encryption, that is, if you don't have HTTPS and TLS encryption, you have no authentication of the endpoint, meaning that anybody could intercept your connection and have at what's going on back and forth.

Also the router interface allows users to enter credentials for their email accounts, which are then sent from the router to the server without encryption or verification. And the passwords are stored in cleartext. So, I mean, the more you look at this, the more it looks like a whole bunch of features that look fine on the surface, but absolutely no attention to underlying security and protocol. Just none.

There's a backdoor on the LAN. There is a secure tunnel both on the WAN and the LAN side, but they use hardcoded private keys which anyone can determine they never change. And so even though you use stunnel in some cases to talk to this thing, the keys are known and fixed, so no actual protection. And nothing like Diffie-Hellman dynamic key exchange. Just no one's caring here.

There's also, I mean, it just goes on and on. There's a way to brute-force the DNS configuration to allow the DNS config to be changed without admin user authentication

checks, which allows anyone to reroute your DNS to a malicious DNS server. And we know that that's like the first thing you want to do if you want to spoof a website's identity is return fraudulent IP addresses for legitimate DNS queries. There's weak file permissions, and credentials are stored in the router in cleartext, which a remote attacker has access to.

There are remote code execution vulnerabilities where you have the ability to run as root; the DHCP server, which we know that a router will [dropout] get its IP address and gateway and other setup details over the Dynamic Host Configuration Protocol. And it turns out that that daemon in the router has all kinds of vulnerabilities that allow that query to be manipulated to allow a remote code execution with root privileges. And there's also the ability to DDoS or DoS some of the daemons that are running in the router.

So D-Link, if you do have an 850L, be very wary. This is one of those situations where just replace it with something like the $49 EdgeRouter X or with a pfSense router or something. It's just, you know, we're seeing these things being sold which, from companies that evidence no interest in the underlying security of their offerings and of the concern for their customers. And end-users don't know. It's up to security researchers and vehicles like this podcast to help spread the word and just to create, I think, a level of awareness that unfortunately has to be present.

It's going to be interesting to see how this evolves over time, how we create more responsibility, except maybe for consumers over time to just wander away from companies whose reputations have not afforded them the kind of demonstration of concern over security that they should have.

So, okay. We've talked about the problem that Symantec had with mis-issuing certificates and, in one case, one of their partners that they had delegated responsibility to not being responsible with the issuing of certificates. And the industry has decided that they are going to sunset trust. So what happened yesterday was that Google's security blog updated the industry on where they stand. And it also included some important information for site owners which I want to cover, and then also bring up something that I don't think we talked about before, which is the potential impact on IoT devices that this would have.

So Google wrote in their blog, they said: "This provides a bit of Google-side perspective about the Symantec/DigiCert agreement and what it means also practically for website operators." They said: "After our agreed-upon proposal was circulated, Symantec announced the selection of DigiCert to run its independently-operated Managed Partner Infrastructure, as well as their intention to sell their PKI (Public Key Infrastructure) business to DigiCert in lieu of building a new trusted infrastructure." And this is what we talked about before is that Symantec just said, okay, we're going to transfer our business unit, essentially, and the public key infrastructure, to an already trusted certificate authority, in this case DigiCert, because that just makes the most sense.

So Google explained for website operators what this means. We were just talking about how this poor man-in-the-middle implementation detection will work in Chrome 63. Starting with Chrome 66, which is around March 15th of 2018 in beta, and the stable track users will get it about a month later on April 17th, 2018. So starting with that version, Chrome 66, Chrome will remove trust in Symantec-issued certificates issued prior to June 1, 2016. So prior to this summer, June 1st of this summer. And so that happens like after the first quarter of next year.

So Google writes: "If you're a site operator with a certificate issued by a Symantec

certificate authority prior to June 1, 2016, then prior to the release of Chrome 66 you will need to replace the existing certificate with a new certificate from any certificate authority trusted by Chrome." So that's, I mean, hopefully all sites who are using such certificates are aware of this and getting this news because it'll come as a rude awakening after the first quarter next year when Chrome users are no longer able to connect to them. They'll be getting errors from Chrome.

Okay. And in addition, by December 1st of this year, so a few months from now, December 1st, 2017, Symantec will transition issuance and operation of publicly trusted certificates to DigiCert's infrastructure, and certificates issued from the old Symantec infrastructure after this date will not be trusted by Chrome. So that says that there is a period between June 1st and December 1st, so through this summer into fall, where certificates issued then are trusted, but Symantec's operations will be shut down and fully transferred to DigiCert by December 1st. So now certificates should be issued by Symantec; and, in any event, Chrome won't trust them because, again, remember, there's just a concern that we're not really sure the extent of what happened with Symantec. Reports have varied. But it's better in this case just to say, okay, we're no longer going to trust those.

And then finally, around the week of October 23rd, 2018, so more than a year from now, about 13 months from now, Chrome 70, seven zero, will be released, which will fully remove trust in Symantec's entire previous infrastructure and all of the certificates that have been issued. So essentially what Chrome will be doing until that time, until October 23rd, will be looking at the not-valid-before date, that is, the date of issuance, and making a judgment about when that certificate was apparently signed, whether or not to trust it. And then 13 months from now, October 23rd, 2018, the Symantec root will be pulled out, will be removed from Chrome so that no certificates anywhere ever that were signed using that root will be considered valid.

So we're seeing sort of this drama unfold in real-time. We've talked about these things before, and this is not certainly easy for the industry. It creates some upheaval. But it's what we have to do because, as we've said, the whole system only functions, I mean, and it functions precariously as it is because we're trusting implicitly the signatures by so many different certificate authorities where any certificate can be signed by any certificate authority. And if that signature is trusted, then the certificate is trusted.

But there is a concern here that could have some impact on consumers, depending upon the design of IoT-style systems, because remember we're all used to thinking in terms of our client-side OSes, which have a massive store, hundreds of trusted certificate authority root certificates, which as I said will be used to trust anything that they sign. But IoT devices, which are inherently lean, may presume the certificate authority of the remote server that they trust. That is, our web browsers, for example, have to trust everybody because we're roaming around the Internet. We want our browsers and our systems to be able to go connect to any server anywhere and authenticate the identity and get a secure connection.

But an IoT device inherently has a different architecture. Many of them, for example webcam, are inherently tied to a specific remote server at a specific domain. And because they're trying to keep their costs down, they're trying to keep their support overhead down and not waste RAM and flash memory and so forth, they may well have a single certificate because they know which certificate authority will always be signing the trusted certificate of the remote site. If the authority was Symantec - and the other problem is these root certs are very long-lived. They've got expirations decades from now. So there may well be IoT devices where - and remember that Symantec was a major CA, historically. I mean, they're the ones I used. They were like the one, back in

the day. And I of course happily moved away over to DigiCert quite some time ago.

But it may well be that there are hardware devices that only trust Symantec certificates. And there will be no more Symantec certificates for them to trust before long. And the idea would be that, because an IoT device's root cert has a multi-decade life, the server side could keep refreshing its certificates every two or three years as required, as we know, yet still be trusted by this single long-lived certificate burned into the firmware of some device. So there's really nothing we as consumers can do.

The upshot is that some devices, if they have a single trusted root, and that root is Symantec, and hopefully they are establishing secure connections rather than just unencrypted and unauthenticated connections, they're going to die next year, which will create an additional unexpected problem. We on our much more dynamic, full-spread, client-side OSes, this won't affect us at all. But it does affect the server operators who have to make sure that the certs they're using remain trusted. But again, I'm sure that news will spread sufficiently.

And speaking of this problem of any trusted signer being able to sign a certificate for any domain, note that pinning certificates creates an exception, which we were talking about with Google's Chrome. There is a - I was going to say an "upcoming standard," but actually it already went into effect. Last Friday, on September 8, 2017, the requirement for certificate authorities to check the new, relatively new, actually it's been around for a while, but we've talked about it before, the so-called CAA, the Certificate Authority Authorization record that is being published by a domain's DNS, went into effect.

So the CAB Forum, the CA Browser Forum, established these guidelines. All certificate authorities starting - they have been able to deploy this technology earlier, but it was mandatory last Friday. And what that meant is that, before a CA, a Certificate Authority, would grant a certificate to an applicant, the certificate authority would query that domain's DNS for a CAA record, which was a newly defined DNS record type, which specifies the certificate authorities that that domain uses to sign its certificates.

So, for example, GRC would have a CAA record that says my certificates come from DigiCert. So the idea would be that other CAs should, before ever issuing anyone a server certificate for GRC, query GRC's DNS and see whether there is a CAA record. If there isn't one, then that means there is no block. That is, okay, this person hasn't said who they buy their certificates from. If there is one, that is, if there is a CAA record which says these are where we get our certificates at this domain, and it does not include that CA, that Certificate Authority that is being asked to issue a certificate, then they must not.

So, okay. This is obviously when, as we've seen, this is not strong protection. But we haven't got a way yet to create strong protection. This is better protection. It's very lightweight. It allows anyone who wants to increase their own domain's security to add one of these records so that any other certificate authority who is responsible and does follow the CAB guidelines, the CA Browser Forum guidelines, will refuse to issue a certificate when they should not issue it because they've not been given permission to explicitly by the presence of one of these CAA records.

So unfortunately, the day after this became mandatory, a curious German researcher went over to Comodo, and he has a long-published CAA record for his domain, stating that Let's Encrypt is the source of his certificates. And everyone knows where this is headed. Comodo happily issued him a certificate in contravention of the CA browser guidelines, which had gone into mandatory effect the day before. And this guy got his certificate, verified it. Since then, many other people have confirmed the same thing, so

it's not just a single source of reporting.

What's odd is that Comodo has a very nice-looking page on their site where they go on at great length in their knowledge base about the CAA record and how it works and what it does. And they've got links to various types of DNS servers and what you need to do. But unfortunately, they're not, I mean, this is not just a single report. This is multiple reports now that have verified that this is happening. And they're just deciding, oh, well, this will be a nice technology. Hopefully they will figure out why they issued that certificate by mistake and start honoring that because, again, this doesn't prevent a certificate authority from maliciously or deliberately ignoring that.

But of course, as this researcher just demonstrated, this is also easily tested because, as it is now, any CA - or as it had been, any CA would happily take your money in return for giving you a certificate. Now that shouldn't happen, if there's a CAA record. And again, this is easily tested. So I think what we'll see is quickly that things come into compliance because we've got serious companies like Google who have demonstrated that they're willing to subject companies to whatever degree of pain is necessary in order to keep the system up and functioning and trusted, which is what we need.

I've got two quick little bits, some follow-ups on some previous coverage of ours. We talked, boy, about a year ago, whenever it was that Volkswagen got caught basically with special case software which was able to detect when its emissions were being tested because the back wheels were spinning, but the front wheels were not. And that sort of just sort of disappeared. It just resurfaced again. It turns out that, unfortunately, one of the engineers who was involved in this, he's in charge of the group that produced the software that did this. He's been given 40 months of jail time...

Leo: Whoa.

Steve: In prison, yeah.

Leo: I think the executives should get the jail time.

Steve: That's my feeling, too. That was my immediate reaction was, wait a minute, you know, this guy was probably following orders. So also he's paying a $200,000 fine. TheRegister.co.uk picked up on this. They said: "As head of the VW Diesel Competence unit in the U.S." - and the engineer's name is James Liang. He "oversaw the software function that enabled the cars to cheat the emissions tests. He is also the most junior of the eight current and former VW executives that have been charged so far."

Leo: Fall guy.

Steve: So, yes, there is some, you know, a bit of a - there is responsibility being taken. They called it the "defeat device," which was designed to recognize when the car was being tested and to switch to a lower emissions setting. When the car was running normally, that setting was removed, and emissions were measured at up to - get this - 40 times higher than the permitted levels.

Leo: Yeah, but the response was great. That car could go.

Steve: And this, of course, was put on 11 million cars. The Register writes that: "The engineers knew full well what they were doing and attempted to hide their tracks, even calling the device by a variety of pseudonyms including 'acoustic function,' 'cycle-beating software,' and 'emissions-tight mode.'" So it turns out that, in terms of, like, sentencing guidelines, basically the judge threw the book at these people.

Leo: Good.

Steve: The federal prosecutor said that Liang's prison sentence would send "a powerful deterrent message to the rest of the industry." And apparently not just the auto industry will hear that message. "Software engineers across the country," write The Register, "will have to reflect on the fact that they may be held personally responsible for creating something that knowingly breaks the law."

Leo: Good.

Steve: So, yes, exactly.

Leo: Although I really want to see the executives go to jail. I don't...

Steve: I agree. I agree completely.

Leo: I'm sure he was just following - "I was just following orders."

Steve: Well, and I guess he's going to say no, and then get - I don't know how it's [crosstalk].

Leo: Well, from now on I think that's the hope is that people will think twice and say no when they're asked to do stuff.

Steve: Yeah, I'm not going to prison for you pencil necks, yeah. One other little bit of news about iOS 11 came across my radar actually last week, but we didn't have time to talk about it, and that was that we'd talked previously about the problem of MAC address leakage from our iPhones, and that Apple had taken some - was aware of the problem, that is, in the Ethernet protocol, we know that MAC addresses are used to carry packets. And MAC addresses are globally unique identifiers by design because the way Ethernet works, that's something that you need to count on not having an address collision with. There was some provision for Apple to obscure the MAC address, that is, if you were probing versus connected, as I recall from our previous discussion of this.

Anyway, what Apple decided to do with iOS 11, and bravo to them, although there are

some packet capture people that are annoyed, and that is they are removing access to that information completely so that you will - apparently apps have previously had access to the ARP table. That's the Address Resolution Protocol table where essentially that's a rich source of information. You're able to see all the MAC addresses and IP addresses of all the machines that you're currently seeing around you in your environment, including your own.

So essentially they're going to just remove that from the application space so that applications will no longer be able to query that, to further increase the privacy. So that's, you know, most applications have no need for it. There are some developers who are chafing at this news because their particular packet-sniffing, packet-capture widget had access to it, so they're not happy that they're losing it, but tough. I think from a privacy standpoint it is underlying plumbing information which definitely can be used to breach privacy, and that's a problem.

I had this in my notes last week, but we didn't have any time. And the news was pretty much focused on Equifax this week, so we have a little more time. I wanted to take a minute to do something I only do every couple of years because this sort of - what I see is that, as I'm talking in passing and sharing testimonials about SpinRite, I start beginning to get email from newer listeners who are saying, well, you talk about SpinRite every week, but you never explain really what it is, just like it does data recovery stuff. And so over the course of the 12-plus years we've done the podcast, every couple years I've said, okay, let me just quickly explain what this is. I also want to talk about briefly, catch everybody up on the 6.x plan, my plan for 7, and also the pre-release for the 6.x stuff.

So for people who don't know, for listeners who are new to the podcast, SpinRite was born more than 30 years ago in arguably a very different era of the PC industry. Hard disk drives were very small, like 10, 20, 25 megabytes, not gigabytes, and not terabytes. And they used much less technology, so there was much more [dropout] placed on the user and on the [dropout]. The drives themselves had no brain in them at all. All the brains were in the controller. And SpinRite started off with the goal of improving the performance of the drive because drives at that time were unable to read the data as fast as it was coming off of the disk. The disk was spinning too fast. The data was already too dense. There wasn't a chance to get it into the computer, into its memory.

So instead there was this interleaving of sectors was a technology that had already been applied in the non-PC industry, where instead of successive sectors being adjacent, successive logical sectors were put downstream. They were interleaved so that, like, every sixth sector would be the next one. That allowed one sector to be read and then some time to pass while that got into RAM so that you could then ask for the next sector to be read. The problem was there was no provision for optimizing that interleave. That is, for example, IBM set theirs to six, that is, a 6:1 interleave. But that meant that it took six revolutions of the drive in order to read all the data from one track, and then you could go to the next one.

What I discovered 30 years ago was that that was not optimal; that many systems could do it in three, that is, a 3:1 interleave, and some cases a 2:1 interleave. What that meant was that you could reduce it from requiring six revolutions in order to suck in the data to just two, so it made your drive three times faster, and that was a big benefit.

The problem was, in order to do that, it required that the drive be low-level reformatted, to actually physically change where the sectors were. Well, that was something you could not do once the drive had data. Except that, back then, nobody had anywhere to put their hard drive data except keep it on drive. So the big innovation that I created back

with SpinRite 1 was the world's first and only non-destructive re-interleave. And I did that by tackling it one track at a time. Turns out you could low-level reformat just one track. And so I would read all the data off it, and then I would change the sector ordering to optimize it for data transfer, and then put all data back.

But being a perfectionist, I thought, okay, I need to make sure I get all the data off because, once I low-level reformat that track, there's no more data there to get off. So part of the responsibility I undertook was to absolutely, positively do everything I could to get any data off, especially if the drive wasn't wanting to give it. You know, drives back then, as I said, didn't have a lot of technology. They had much lower levels of error correction. They didn't depend upon error correction nearly to the degree we do now, which has happened as bit density has increased.

So I built into the very first version of SpinRite arguably the world's best data recovery as a side effect of the fact that this was going to be my last opportunity ever to retrieve data from the hard drive before reformatting it at the lowest level and then putting the data back. So consequently, SpinRite always had world-class data recovery.

So years go by. Drives get more dense, much more data on them. And at some point the controller moves into the drive with what we called the IDE drives, and SCSI drives were also very much the same way. They became smart drives and began to encapsulate more and more technology. Of course there was also huge pressure to squeeze as much data onto the drives as possible. And as we've talked about, there were much more technology applied in order to always keep the drive kind of on the verge of working, where now more error correction was being used, almost continuously, but much more capable error correction also, in order to deal with the inherent softness.

So today's SpinRite no longer re-interleaves drives. I can't remember where I took that out. I think I took it out when I went to SpinRite 5. I said, okay, I mean, just we're not doing that anymore. All drives are now 1:1 interleave. The channels from the drive to the processor and to main memory are all fast enough to handle taking the entire track off in a single revolution. So SpinRite has evolved with the industry over time, no longer optimizing sector interleave, but actually falling back on and becoming useful going forward.

And amazingly, even as we've spoken of, on SSD drives that don't spin, but do use error correction, which is where SpinRite still comes in, it's evolved into a powerful data recovery system. Which is what we talk about when people's files are in danger, run SpinRite on it. Drives die, run SpinRite on them, bring them back to life. You have to make a value decision, how much you want to trust it after it died once because, if it's trying to die, and I've said this before, ultimately SpinRite will fail because if the drive is determined to stop being a hard drive at all, it gets the final say. But SpinRite certainly pulls you back out of a gray zone and keeps your data there typically long enough for you to then back it up or image it or get another drive and copy it over.

So we've been at SpinRite 6.0 since 2004. It's now 13 years. And it's [dropout] need to update it. That's what I will immediately return to as soon as SQRL is behind me, and that's looking really good. We're struggling with a couple last details, but we're getting very close. So my plan is to - ultimately I want to go to SpinRite 7 and offer a whole bunch of new features because SpinRite is still just sort of a run-it-in-place system. It made sense back then, but now people have drives everywhere.

So there are a lot of other features I want to add. But I'm going to use the 6.x series - 6.1, 6.2, 6.3 - as sort of a technology development platform to create the technological foundation for 7. So 7 will basically be a complete rewrite, but using the foundation that I

develop with 6.1, 6.2, and so forth.

So what 6.1 will get us is a final removal of the BIOS and BIOS dependence, which has been a compatibility benefit in the long term, but recently a problem because BIOSes have not kept up with the drive technology. So that'll go away. SpinRite will talk directly to the hardware. And I had all that running when I suspended the work on 6.1 in order to develop SQRL. I'll go back to that. That's how I know how fast SpinRite was running, and we've talked about that technology before.

But the idea will be the 6.x series will create this new technological foundation that will be useful immediately for 6 owners and will create the foundation for 7 moving forward. And I will, as I have committed, make all of the 6 series available at no charge to all existing SpinRite 6 owners. And because I want to give a thanks to the listeners who have been supporting me and making this possible by purchasing SpinRite 6, it will also be available to us on a pre-release basis. That is, that always tends to be the way things work out.

Like SQRL has actually been working, not in finished form, but working for two years, or a year and a half at least. And Leo, you'll remember a long time ago I looked at the screen with SQRL and logged you in over our Skype connection.

Leo: I do, yeah.

Steve: So I'm sure the same thing will happen. SpinRite will be running, that is, [dropout], well before I'm able to say, okay, I've got all the UI, or the packaging for the typical end-user. So I'm going to make that available early to the podcast's listeners as a special thanks to everybody.

Leo: Very nice, thank you. Well, somebody just joined the show, and he said, "Have you talked about Equifax yet?" That's next. Can't wait to hear Steve's take on EquiF'd is what we call it. You've been "EquiF'd."

Steve: I was tempted to call it "Equihax," but I thought, nah.

Leo: Equihax, yeah, I like it. Yeah, no, it's good. We're going to get into it in a second. Steve, I'm not going to be here for the next couple of weeks.

Steve: I know.

Leo: I think Father Robert is going to be taking the helm, depending on whether he's got work for the Pope or not. If not, it'll be Jason Howell.

Steve: And you're not kidding when you say that.

Leo: And I am not kidding. But Robert loves doing this show, and I know you love

doing it with him, so I think it'll be Robert, Father Robert Ballecer in the next couple of weeks. I'll be back in October.

Steve: And where are you going?

Leo: France.

Steve: Oh, nice. So I did have one closing-the-loop item. An Ed Zucker asked what do I think about Tarsnap as a replacement for CrashPlan? And I've spoken a little bit about Tarsnap in the past. The short version is absolutely yes. It was designed and created by a very good guy, Colin Percival. And it's T-A-R-S-N-A-P dot com. It runs on Unix-like OSes, so not for Windows, although it can run under Cygwin. So the BSDs, Linux, macOS, and so forth. And it's security done right. It's full TNO, and it's a service as opposed to a standalone product. So Colin hosts the cloud side, but it's very affordable. And he shows you how to compute your costs, and things are measured in picodollars.

Leo: I love it, 250 picodollars per byte-month.

Steve: Exactly.

Leo: It's hysterical.

Steve: Yeah, he's a neat guy. In fact, he's the one who designed the scrypt memory-hard function for Tarsnap which I took and created EnScrypt for use with SQRL. So that gives you a sense for how much I respect his crypto stuff. He has a very correctly implemented block deduplication system. You can't do deduplication after you encrypt. So he maintains a hash of blocks on your local client. And then, when you're doing an update, new blocks are hashed and checked against the cache to see whether it's already been stored. If so, it doesn't do it again. If not, then it hashes the block, adds it to the hash cache, then encrypts it locally and sends - oh, I'm sorry. It then compresses it and then encrypts it and then sends it up to the server. So if anyone is looking for a replacement to a cloud-based system, who has a Unix-like OS - BSD, Linux, macOS and so forth - Tarsnap I can recommend without hesitation because Colin definitely knows his technology.

Okay. So a listener wrote, he said, actually tweeted: "@SGgrc, not an American so I don't really understand what Equifax does. Mayhaps there'll be an explanation on this week's Security Now!." Let alone the title.

Leo: There must be something. There's no way, I mean, I'm not a fan of the credit reporting agencies, but I can't believe that in Amsterdam you would just walk in and say I need a car loan, and they go okay. They're going to check your credit.

Steve: Don't know how they do that.

Leo: There must be some way to do it. They just don't call them the same thing. But there's got to be.

Steve: Ah, maybe that's it.

Leo: Yeah.

Steve: So for our listeners who don't know, what has sort of evolved in the states, in the U.S., is a symbiotic relationship between credit grantors and these credit bureaus, as they're called. We have three of them: Equifax, Experian, and TransUnion. And so the idea is that someone who's going to grant you credit, a bank or a credit card company, wants to have some warm and fuzzies about your history of paying your debts. Are you good for the loan that they're going to give you in one form or another?

So these credit agencies, they aggregate the history of individual U.S. consumers, and apparently not only just U.S. There's some Canadian and I think some U.K. that somehow get caught up in this. So they maintain this history. And so there's sort of, I guess, a Hobbesian bargain going on because the credit grantors say, well, we want the ability to know the history of people we don't already know. And so these aggregators say, well, we'll tell you the history of people you ask us about in return for you telling us the delinquency or failure to honor the credit, oh, and their credit lines and everything else, of the customers you already have. So there's this bidirectional flow of information.

What ends up happening is that these three agencies aggregate over time a huge amount of very personal data and background. In fact, in order to do their job they're a database that has to store detailed information, private information about consumers. And notice that this is not anything that a consumer asks for. I never signed up with these companies, yet they've got complete records of my entire credit history without me doing anything, without me giving them permission. This is all from - what?

Leo: Well, you don't give them permission. But whenever you acquire a loan or buy a car or rent an apartment or get a credit card, in all of that documentation you give permission for them to report all of your activity to a credit reporting agency.

Steve: Right.

Leo: So you don't give it to them specifically, but you've agreed to it.

Steve: It's in the fine print, essentially.

Leo: In the very, very fine print, yeah.

Steve: Well, and the other thing I'm seeing, and I'm sure you are, too, Leo, is from time to time you get letters from your bank or from investment firms or whatever who have [dropout] saying, you know, we're updating our privacy terms. And if you don't do

anything, then you're implicitly giving us permission to share your information with others, with our marketing partners or our financial partners. And so unfortunately a U.S. consumer has to be very proactive if they want to minimize the amount of information which is leaked about them and hungrily aggregated by these companies.

And of course what is then evolved is the so-called "credit score," which reduces all of this through some sort of equation to a number. I don't know, is it like zero to a thousand or zero to...

Leo: 850, something like that, yeah.

Steve: 850? Okay, yeah.

Leo: Maybe it's 900. I've never seen 900, but I guess it's theoretically possible.

Steve: And I did, coincidentally, I was talking to someone the other day who was having a problem, and her score was like in the 400s...

Leo: Yeah, that's pretty low, yeah.

Steve: ...because she was considered not a very good risk. Okay. So what happened? Equifax lost control of this database. Now, and again, we don't have complete disclosure yet. I'm sure Congress is going to be getting complete disclosure because, I mean, this was, in some cases, I know that I saw a piece that Dan Goodin wrote for Ars Technica, wondering whether this might not be the worst personal information disclosure so far in history.

I mean, it's bad: 143 million U.S. customers, which is approximately 44% of the U.S. population, had highly personal data, including of course the names, but also their Social Security numbers, their dates of birth, their physical addresses, in some instances their driver's license numbers. And a subset, about 209,000 also had their existing, their current credit card account numbers; and there were also some sort of dispute documents where you go back and forth in order to say, wait a minute, you've got some bad information about me. And I've had that happen. I've never fought them. But, for example, they had some bogus physical addresses for me in one of them, I don't remember who it was. But it was like, okay, I never lived in Montana ever. But they seemed to think that was a previous address for me.

So there is, unfortunately, they're sucking all this stuff up, and there's no feedback unless you go and pull your report from these people and then argue with them about things that they got wrong because otherwise it's just sort of being done passively. Oh, yes, and information also about U.K. and Canadian residents were also there.

So what we know from still kind of fragmentary information, but I've now found it in multiple places, and in some cases financial ratings, that is, stock market equity summaries that are very careful about being correct with their facts. We believe that Equifax first became aware of the incident on July 29th - so, what, five weeks ago, six weeks ago, okay - while the breach is believed to have occurred somewhere around mid-May, meaning that half of May, all of June, and all of July, two and a half months before

they were even aware.

So that's raised throughout the industry a big red flag. It's like, whoa, wait a minute, this company that we're trusting with this kind of information could have somebody accessing their data - and apparently this didn't happen all at once, it was over time - for two and a half months, undetected, until they figured it out. But then, of course, once they did, on July 29th - so this only came to light last week, which means [dropout] five weeks after knowing that they had had a catastrophic breach in their security affecting 143 million U.S. consumers, putting our credit at risk. Because the idea is that, with this information that they had that got loose, other people could impersonate us and apply for loans and credit under our name and get credit cards and change addresses and so forth, and thus really cause havoc.

So it's also a concern, not only that they didn't know about this for two and a half months, but that they waited for five weeks before letting us know. And there was also some reporting that tried to make a deal or make an issue of the fact that three Equifax executives cashed out $1.8 million of their stock during this interval, but there has been a formal statement from Equifax saying that none of those three executives had any idea. They were not in the loop, so this was not them trying to cash out before Equifax's stock crashed. And no one's expecting this is the end of Equifax. The one equity firm whose report I read said, well, maybe it'll take a 10% hit, but they'll recover over time. It's not the end of the world. Still, it's disturbing that this is the way they behaved.

So, okay. What do we know about how this happened? That was sort of what happened. Well, what we know is that the Equifax system is based on a well-known open source server-side Java-based web design framework called Apache Struts. It's a sophisticated system - that is, Apache Struts is - that needs to be used responsibly. You know, Java is a serious industrial-strength language, but it's not a toy. You need to know how to use it correctly. So it doesn't do a lot of handholding for you, requires competent developers. And as we'll see in a bit, when we talk about the generation of the per-user unlock authentication PINs, the competence and attention to detail of Equifax developers easily calls their competence into question.

So because Apache Struts and the Apache Struts project, which is part of the Apache Foundation, has been called into question, Ren Gielen, who is the VP of the Apache Struts Project Management Committee, went on the record to address the fact that they seem to have been implicated, their code appears to have been implicated in this. And there's some interesting information here. So he said: "The Apache Struts Project Management Committee would like to comment on the Equifax security breach, its relation to the Apache Struts Web Framework, and associated media coverage.

"We are sorry [of course] to hear news that Equifax suffered from a security breach and information disclosure incident that was potentially carried out by exploiting a vulnerability in the Apache Struts Web Framework. At this point in time it is not" - and I edited this a little bit. "At this point in time it is not clear which Struts vulnerability would have been utilized, if any. In an online article published in Quartz [qz.com], the assumption was made that the breach could be related to" - and there's a CVE number, we'll call it the "9805." It's 2017, so it's this year, 9805, which was publicly announced eight days ago on September 4th, along with new Struts Framework software releases to patch this and other vulnerabilities. So Apache Struts is being immediately responsive and responsible.

However, they write, the security breach was detected by Equifax in July, which means that the attackers either [dropout] an earlier announced vulnerability on an unpatched Equifax server, okay, so that would say that there may have been something else that

was previously known and patched in the open source community which Equifax may not have bothered to keep current. That's a theory. We don't know that yet. Or they exploited a vulnerability which was not known at the time, thus they may have independently discovered this 9805 known problem that was disclosed last week, or something else. Again, we don't know. No doubt there's forensics people looking at this now to get a better sense for what was going on.

Frankly, given what we will talk about in a second about the construction of the PIN, the secret passcode that users are given when they lock down access to their credit reports, it really doesn't inspire much confidence. Not to mention the fact that they are explaining about how to use Netscape Navigator on their site. So if the breach was caused, they write, by exploiting this 9805 known, now known as of last week, it would have been a zero-day exploit at that time.

And the Quartz [qz.com] article also states, and these guys didn't respond to that, probably because they're embarrassed, that this 9805 vulnerability has existed for nine years. So it's been there, like, forever in Internet time. And if it were independently found, that would be a problem. But again, as we know, mistakes happen, and Apache Struts immediately fixed it and issued an update. We don't know what Equifax was using on their servers and whether they were being as responsible or not.

Okay. So this goes on for a while, but it ends with properly phrased good security advice, just to sort of wrap things up, which was keep your frameworks and libraries up to date. Make sure that you're staying current, that is, that you're using the latest that is available. A note that complex software always contains flaws. And so don't depend, as Apache Struts guys, don't build your security policy around the assumption that the supporting software products are flawless, but rather use a defense-in-depth approach, a multilayered approach.

And, importantly, monitor. As we've said often, it is ultimately monitoring to see if what's going on on your network makes sense. If you can explain all of the traffic that you're seeing, that's crucial. No matter what your defenses have, you also have to watch. So anyway, so that's the techie open source web framework side of this.

So finally, the consequence in the public is, not surprisingly, that in the wake of this news, everyone has been rushing to lock down their reporting. As I said, the danger to consumers is that a huge database, 44% of U.S. consumers have enough information now loose, including their Social Security number, in some cases credit card numbers, their date of birth, their physical address, I mean, everything you need, typically you provide to a credit grantor to grant you credit, is now available.

So our listeners who followed our advice many years ago, two and a half years ago, may already have been protected. On Security Now! Episode 495, which you and I, Leo, recorded on February 17th, 2015, that podcast was titled "HTTP/2." And I said, and Elaine transcribed, I said, "Okay. Lastly, and this is - I probably should have done this first, but these other stories were just too interesting, and the industry's been buzzing about them. I created a bit.ly link for this, and it's important: bit.ly/freezecredit, all lowercase. This takes you to a page where the guy explains that a service that is available to anyone for all three major credit bureaus - Equifax, Experian, and TransUnion - allows consumers who are not the victims of identity theft to lock their credit reports. It's not free, but it's not too expensive. The cost varies..."

Leo: This is Clark Howard's page. It's a good page, yeah.

Steve: Yes, yes. "The cost varies, depending on where you are, from $3 to $10. In California it was $10. And because I was curious to do this, although I wasn't worried about this recent Anthem breach" - and that's what brought this current was it was the Anthem breach two and a half years ago. I said: "I locked my credit reports. It was $10 for each." And you, Leo, responded: "You may have to pay to thaw it, as well, and that's important."

So we talked about this two and a half years ago. I, my friends, and family locked ourselves down. So what that meant was that it is impossible, given that the lock is honored, impossible for anyone to apply to use these three firms' reports in order to get credit granted. And so, for example, if I needed a loan for something, I would have to explicitly unlock access in order to permit it.

And of course there's been a lot of controversy about this, naturally, in the last week because this has put a big spotlight on the whole locking/unlocking process. And many consumers are annoyed that they're being charged to do something that they never asked for in the first place. But in truth, and as you said, Leo, in order to be granted credit, you need to prove your creditworthiness. So this is indirectly a service for consumers, although we certainly need the people who maintain all this data to be responsible with it.

So, weak PINs. Someone tweeted to me: "@SGgrc How much entropy is there in a timestamp? Would be great to hear the breakdown in tomorrow's show." The answer to the question, how much entropy is there in a timestamp: none. Zero. It is entirely deterministic and predictable [dropout] about a timestamp. It increments uniformly. I mean, it's like none.

Okay. So Equifax said, in response to people noticing that the PIN was a timestamp, they responded: "While we have confidence in the current system" - okay, stop right there. What? They said: "...we understand and appreciate that consumers have questions about how PINs are" - geez - "how PINs are currently generated. We are engaged in a process that will provide consumers a randomly generated PIN."

Leo: No.

Steve: What an innovation, Leo.

Leo: No.

Steve: Imagine that. "We expect this change to be effective within 24 hours. A consumer has an option, and will continue to have an option..."

Leo: Such idiots.

Steve: Oh, Leo. Get this. The PIN is mmddyyhhmm - month, day, year, hour, minutes. And when I saw that...

Leo: Couldn't possibly be a collision.

Steve: Well, could you guess it? No, you couldn't possibly guess it.

Leo: No, never. Never in a million years.

Steve: I went back and looked, and now I'm a little annoyed with myself for not actually looking at the PIN.

Leo: You didn't notice it, yeah.

Steve: Yes. But I know the day, date, and hour, and minute that...

Leo: That you applied.

Steve: Because it's right there in my record of my, quote - you know. Oh, my lord.

Leo: So terrible.

Steve: Yeah. Again, so this is what I was saying. If a company is issuing you a secret token which is a time and date stamp of when you submitted your request online, it's no security. And it is obviously brute-forceable. It's just - it's incredible. So in the best case it's very low entropy. And I can't explain this.

I mean, we already know you could take a counter and hash it and give somebody eight digits or 12 digits or whatever you wanted of that as a PIN. Or if you wanted to be able to relate it [dropout] take the timestamp with a nonce, because you want to have a nonce, and symmetrically encrypt it with a secret key. That would give the consumer something unpredictable and completely random. But if they gave it back to you, you could decrypt it back into that original timestamp, if for some reason you needed a timestamp. I mean, it's just like it's so obvious how to do this right. And these people just thought, why bother? So again, it's difficult to give them latitude.

And of course, finally, we have the expected class-action lawsuit arising from this. A couple of plaintiffs who were tweaked by this found themselves a class-action firm. I'm sure they were lining up to go after Equifax. And so there's a class-action in the works seeking as much as $70 billion in damages for all of the consumers that were affected by this. Okay. So the bad news is that, in the wake of this disclosure of this disclosure, consumers did, I mean, like my best buddy Mark has been frantic on the phone with me, "What do I do, what do I do?" I said, you know, I'll have a full readout by the end of the podcast. I'll be up to speed, and I'll have dug into this.

Everybody has been running to lock their reports down, as we discussed two and a half years ago. And for what it's worth, again, that bit.ly link, bit.ly/freezecredit, is still valid. You just brought the page up, Leo, and it's a good page to talk people through how to do

that. The problem is that the three bureaus have been crashing because of the demand on their sites for everyone who wants to pay $10 or whatever it is, $10 in California, to each of the three agencies.

Leo: In some states it's free. In Maine it's free.

Steve: Ah, good.

Leo: So don't assume it's going to cost you anything. It's just, you know, each state has its own rules on that.

Steve: Yeah. And in fact in our reporting two and a half years ago I said between $3 and $10. So, but again, zero and 10. Probably as a function of what the state allows these guys to charge. Maine probably just said, no, you're not charging anybody anything for that.

Leo: Exactly, yeah.

Steve: Because they would certainly like to, if they could.

Leo: Oh, yeah.

Steve: So the final takeaway is, depending upon how concerned you are, remember that there's an inconvenience associated with this. I've been locked for two and a half years because I haven't needed to apply for credit in two and a half years. I like my credit cards. They're fine. And so I'm at a stage in my life where I'm not buying things that I need loans for any longer. So that lock I applied, I mean, I know exactly when because my PIN tells me, back when I was researching this after the Anthem breach. But it is an inconvenience if you need to be unlocking it all the time, especially if they're going to ding you again for the privilege of releasing the lock on your account, and then you're going to want to go back and re-stealth yourself by locking it again.

The system's a little broken. But unfortunately it's an asset of some value because it does allow people whose credit has been established to have the freedom of having a third party to represent that, yes, you've been making your payments on time. But with it comes a responsibility that Equifax, I mean, at worst they made a mistake. I think maybe, as more facts come out and this unfolds, we'll get a better sense for just how irresponsible they were. We don't really have a calibration on that yet.

Leo: Amazing.

Steve: Yeah.

Leo: There is, you know, we talked, didn't we, about that chatbot that you could use to fight traffic tickets in the U.K., and they launched it in the U.S. Apparently he's added a feature now that you can sue Equifax using that. I think it's for like $35,000 using the chatbot. So I think Equifax is going to get some heat for that, quite a bit of heat from this. Congress is investigating.

Steve: Yeah, there is an automated - there's something that I missed, also, Leo. I don't know if you picked up on this. But there was something about something that you did which caused you in the fine print to lose the ability to sue them.

Leo: Yeah. That's actually not the case. Well, it might be the case.

Steve: I didn't think so, and that's why I didn't cover it was I was never quite clear on what it was.

Leo: So if you went, and no one should ever do this because no reason to deal with Equifax any further. But if you decided to go to their goofball site where you would figure out if you'd been hacked, Brian Krebs went there and entered random name and numbers and got the same responses if you didn't. So it's a goofball site. And apparently its real purpose is to move you through to their free credit monitoring TrustedID. If you agree to that, there is an arbitration clause in that.

But both, first of all, I don't know how binding it would be. Secondly, Equifax has said, well, this only applies to our TrustedID service. It does not apply to the breach. We're not going to assert that you can't sue us. Although they've been fighting hard. They've been giving lots of money to members of Congress to eliminate the ability for consumers to sue and eliminate the CFPB, the Consumer Finance Protection Bureau.

Steve: God help us.

Leo: I mean, these guys are as bad as it can get. They're just as bad. They collect, I mean, we talk about Google and Facebook invading privacy. These guys do it. They collect all your data, they don't secure it, and then have the gall to try to make money off of this breach by signing you up for free protection which will of course on year one plus a day start charging you. It's just - it's kind of - the gall of these guys is mindboggling. Mindboggling. I would like to know what they do in other countries for this kind of credit reporting. They must have some system. Maybe it's run by the government? I don't know. But there must be, there has to be a way to verify your creditworthiness.

Steve: [Dropout] free enterprise.

Leo: Well, that's the thing. We live in a capitalist society. And this is not only legal, it's kind of encouraged and supported by the government. Well, Steve, what a good

job you've done of synopsizing this. And you warned us all a long time ago, freeze your credit. It's kind of a pain. If you're young, and you're doing a lot of credit, you know, buying stuff, getting credit cards, getting set up in life, it may not be worth the cost and the hassle of freezing and unfreezing. But us old farts should definitely do it. Besides, we've got more to lose. We've got assets. What's your opinion on credit monitoring? Have you ever kind of thought about that or delved into that?

Steve: Yeah, and that was another question that my buddy Mark asked. He said, "What about LifeLock?" And I said, well, when the FTC stops suing them for overstating their benefits to the consumer, maybe. I don't know. I mean...

Leo: There's a story with LifeLock. I'm not convinced that that's LifeLock, by the way.

Steve: Ah.

Leo: I should mention that I think they have been in the past a sponsor of some of the shows, maybe the radio show. But I use them and pay for them. The reason LifeLock got sued, and it didn't get sued by the feds, but it got sued by many states - I think the FTC might have ended up going after them - is that Experian, TransUnion, and Equifax went after them because what LifeLock originally did was put credit freezes on your account for you and maintain them.

Steve: Ah. Nice.

Leo: And that means that those companies can't - Equifax, TransUnion, and Experian can't make money off of you.

Steve: Can't sell your data. Right.

Leo: So they went after these guys. And what they did, and we've seen this methodology used before, they went to individual states attorneys general and said, you know, it would be nice - here's a little campaign contribution - if you were to go after these guys. So it's kind of too bad because LifeLock ended up having to stop doing the credit freezes. They succeeded. But what they did was I thought quite smart. They bought one of the big backend companies that does the transactions for most credit card companies. And this is the company that does the fraud alerts for your credit card company. So they do have access to a huge data flow.

But even then the attorneys general and the FTC said, well, you can't say you can watch all transactions. Well, obviously. Cash transactions don't go through these services. There's stuff that happens you can't see. The bigger question is if credit monitoring does you any good. I mean, I have LifeLock, and I get periodically, oh, we just saw your driver's license on the dark web. Well, what am I supposed to do about that now? Nothing. There's nothing I can do. How do you act on it?

Steve: Yeah. And we've all heard these horror stories about identity theft. I mean, how it's almost impossible, like it just ruins your life to have your identity stolen.

Leo: That's what these companies offer, by the way, mostly, certainly what LifeLock offers is insurance and help in restoring your identity. Maybe that's what you're really paying for. I don't know. I pay for it, and I get it for my kids. When this happened I got it for Lisa and Michael, only because why take a chance; you know? Because it's a lot cheaper to buy that than it is to fight identity theft.

Steve: Yeah.

Leo: Anyway, yeah, I think the jury's out on whether LifeLock really did anything wrong, or if this was just the Big Three trying to put them out of business. By the way, there's a fourth called Innovis that is not on this Clark Howard page. But you can go to Innovis and freeze your credit there, too. Interesting, huh. What a world we live in.

Steve's show is available in a fine, high-quality 64Kb MP3 on his website, GRC.com, as are those fine transcriptions Elaine Farris writes for us. You'll also find SpinRite, the world's best hard drive maintenance and recovery utility there and many, many, many free things, GRC.com, including ShieldsUP!, SQRL, his supplement history, all that supplement info.

Steve: The Healthy Sleep Formula.

Leo: The Healthy Sleep Formula. All you've got to do is go to GRC.com, Gibson Research Corporation. And as for us, we have it at TWiT.tv/sn, Security Now!. And we also have video should you wish to see the growth of Steve's moustache over the years.

Steve: As it returns to the world.

Leo: As it returns to the world. And you can also subscribe in whatever podcatcher you use; you'll find Security Now!. Any show that's been around 13 years is probably in every directory. Please do subscribe. That way you'll get every episode. I won't be here next Tuesday at 1:30 Pacific, 4:30 Eastern, 20:30 UTC; but Father Robert Ballecer will. And Steve Gibson certainly will. He never gets sick. He never misses a day. He's just the Iron Man of security podcasts. Next Tuesday we'll see you. Thank you, Steve. We'll see you next time on Security Now!.

Steve: Thanks, Leo, and we'll see you in three weeks.

Leo: See you later, yeah.

Steve: All tanned, or whatever you are after that.

Leo: I don't think it's going to be one of those tanning. I might be a little fatter. Might have my liver force fed. But other than that, yeah.

Steve: Okay, my friend.