GPU RAM Image Leakage

Description: This week we discuss another Lenovo Superfish-style local security certificate screw-up; several new, large and high-profile secure breach incidents and what they mean for us; the inevitable evolution of exploitation of publicly exposed UPnP router services; and the emergence of "Printer Spam." How well does ransomware pay? We have an idea now. We talk about two iOS scam apps, a false positive Bing warning, progress on the DNS over HTTPS front, and rumors that Microsoft is abandoning their EdgeHTML engine in favor of Chromium. We also have a bit of miscellany, news of a cybersecurity-related Humble Book Bundle just in time for Christmas, and a bit of closing-the-loop feedback. Then we discuss some new research that reveals that it's possible to recover pieces of web browser page images that have been previously viewed.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-692.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-692-lq.mp3

SHOW TEASE: It's time for Security Now! with Steve Gibson. I'm Jason Howell filling in for Leo, who is out this week. Steve's going to talk about a whole bunch. This has been a busy week for security. Obviously the huge Marriott and Quora breaches. Also Lenovo Superfish strikes again, but this time with Sennheiser's own version of it. Gorilla printer marketing, thanks to rabid PewDiePie fans. That might not make a whole lot of sense now, but I promise it will. And Steve explores the insecurity of GPU memory. All that and more coming up next on Security Now!.

JASON HOWELL: This is Security Now! with Steve Gibson, Episode 692, recorded Tuesday, December 4th, 2018: GPU RAM Image Leakage.

It's time for Security Now!, the show where we talk about all the latest security news happenings and a whole lot more diving deep. I'm Jason Howell, filling in for Leo. Diving deep because of this man right here, Steve Gibson. How you doing, Steve?

Steve Gibson: Hey, Jason, great to be with you this week while Leo is back on the East Coast as we open December.

JASON: Absolutely. I kind of can't believe that it's December. We were talking before the show that 2018 just as a year for security, about how scary security can be at times. And it really seems, especially looking at today's rundown, that doesn't seem to be anything that has like an end in sight. It's just going to get more and more complicated as we go along. I've just got to get used to it.

Steve: I ought to point our listeners, who are obviously interested in security, to last week's Triangulation podcast, which you recorded with our often-mentioned Bruce

Schneier and was all about his book. And it was fun to sort of listen to it because it sounds a lot like this podcast. And one of the things that he pointed out is that, as we've often said here, that one of the root causes of the problems we're seeing with security is that it's often an afterthought that security is really expensive. I mean, it's not free. Because of the fact that security is the weakest link sort of problem, every link has to be strong because any one of them being weak gives you an opportunity to breach the system.

So what that means is that, in order for every possible link in a chain of dependency to be strong, you have to focus on security for all of those. And it raises the cost. And he made, I think, what was a very good point, which was that people say they want security. And it's like, yeah. Yeah, I want security. And then you ask them, okay, how much are you willing to pay for it? It's like, what? Uh, well, let's just hope that nothing bad happens. And of course we know how that turns out. This is Episode 692 of the result of hope.

JASON: Yes. This is what hope gets you, 692 episodes of Security Now!. People are like, okay, cool. Give me more. Yeah, it is interesting. He talks a lot about regulation as well in the interview. That's kind of the second part of his book is just like, this problem is too big, obviously, for the companies themselves to make these changes and to prioritize it. And so maybe regulation is needed.

Steve: Yes, they would rather it were somebody else's problem. And, well, and he also made the point, we will be talking about the massive Marriott breach as one of our news items. He mentioned that also last Friday. And he made the point that one of the problems is it isn't clear that breaches are truly expensive for companies. Microsoft Research, the research branch of Microsoft, wrote a paper years ago titled something like "The Rational Lack of Concern Over Security." It was a title like that.

And it was weird, but they made the argument and supported it that people are rational actors. And, for example, and it was written back in sort of the pre-strong password/password manager days, where they were saying, look, yes, something bad could happen. But the cost of having a different long password for every site you visit and the overhead of managing it and all of that mess is such that people acting out of their own rational self-interest will still choose "monkey" as their password. Or you can't anymore. You're not allowed to. The website says no, no. But that was Leo's password for years. Worked just fine, and he didn't have to worry about what it was. It was monkey until the bar really changed.

But the point being that it's not the case that people are irrational. It's that we haven't made it easy enough and practical enough to be secure that that's the rational choice. And so this is what we get. And so, for example, yeah, Marriott suffered a 500 million user data breach through a property acquisition they made two years ago, a breach that we'll be talking about in a second, that predated their acquisition of the Starwood properties by another two years. And it's like, oops, sorry. And they'll move forward. They'll still be Marriott.

JASON: Yeah.

Steve: So anyway, this week we're going to talk about some interesting new research out of Korea, South Korea, about another sort of stone that's been turned over and what was found underneath. And so the title of today's podcast is GPU RAM Image Leakage. It turns out that GPU RAM, of which there's ample because images are big, isn't being managed well from a security standpoint. OS RAM is. We all understand. Oh, you know, you've got to be careful about your OS RAM because you could have secrets in it, and so you don't want other processes running in the same machine to have access to another

process's RAM. That's what memory protection, memory management is all about. That doesn't exist in the GPU. Nobody ever really thought about that. These guys said, hmm, I wonder what's in there? And they found out. So we're going to talk about that.

But whereas last week was like this post-Thanksgiving security dead zone, I don't know what happened, but the point was nothing much happened. We still had a good podcast. But this week, whoo, we're making up for that. We've got another Lenovo Superfish-style local security certificate screw-up that I'll remind our listeners who maybe have joined us since that all happened with Lenovo, what that was about, but then also what has happened recently. We've got several new large and high-profile security breach incidents, of which the Marriott is only one. And I want to talk a little about what that means for us, sort of in aggregate.

We've got the inevitable evolution of exploitation of publicly exposed UPnP router services. We've been talking about this happening; and, sure enough, those chickens came home to roost. The emergence of, believe it or not, printer spam. Also some indication of how well ransomware pays. As a consequence of the U.S. Treasury going after a couple ransomware culprits, we now have a sense for that. We also have the story of two iOS scam apps, a false-positive warning that Bing was producing, some very encouraging progress of running DNS over HTTPS rather than UDP where it's all exposed. Mozilla has been experimenting with that with great results.

Also the rumor that Microsoft, believe it or not, is abandoning all of the work on and their Edge HTML engine in favor of, believe it or not, Chromium. We also have a bit of miscellany, a new cybersecurity-related Humble Book Bundle just in time for Christmas that has some tasty goodies on it. We have a bit of closing-the-loop feedback. And then we'll discuss this interesting new research from some people who took a peek into GPU RAM, wondering what they would find, and realized it was possible to determine what web pages people had been visiting, even though that's a no-no from a privacy standpoint. So I think another great podcast for our listeners.

JASON: Absolutely jam-packed. And I knew it was going to be, especially because of the Marriott and the Quora news today. I was like, oh, boy. We've got a lot of ammo. There's just so much more in here. We're going to talk about that. All right. We start with a picture, don't we.

Steve: We do. Now, the reason is there's a little back story here that you wouldn't be aware of. Last week's Picture of the Week was a truly hysterical wiring closet which, I mean, it was just the wiring closet from hell. It was, I mean, wires were crossing back and forth; and, I mean, it was a disaster. And it really generated a lot of feedback from our listeners, who got a kick out of it. Many of them shared some of their own favorite pictures of similar closets. And this one stood out from all of them. I gave this one the caption: "Shelves? Who needs shelves?" Just because...

JASON: Where we're going, we don't need shelves.

Steve: Where we're going, we don't need - exactly. And because this is a much more modest event. This was some church networking closet of some sort. You can sort of in the upper right is a Panasonic, so that's probably the church's phone system. But what I got a kick out of was that there's a silver D-Link consumer router that is just suspended in midair from one of its network cables.

JASON: Oh, that poor cable.

Steve: And it's got, like, what, yeah, exactly, it looks like it has about eight of them. Like all of his little ports are full. But it's just sort of floating there in the middle of the space

saying, yeah, you know, we don't need a shelf. I mean, and there is a shelf right down below. But that network cable coming from above won't let it quite get down to the shelf, so it's just floating in midair.

JASON: Too bad there's no way to extend that network cable. There's just no way. This is the only option.

Steve: Isn't that a shame? Isn't that a shame that they only come in one size, and that one was too short.

JASON: I don't know if I've ever done that exactly. But I will say that I looked at this image, and a part of me could totally identify with it. I was like, yeah. I mean, you know, it's in a room that you never go in. And, I mean, it's probably going to work. So, okay, fine.

Steve: Well, and you can imagine, too, that when this was lashed together, the person doing it probably had the best of intentions.

JASON: Sure.

Steve: It was probably, okay, this is just...

JASON: Temporary.

Steve: ...till I get to Fry's and pick up a long cable. I'm definitely going to do that real soon now.

JASON: And then they realized, maybe I don't need to. It's still working.

Steve: There actually is dust on the D-Link router. You can sort of see some fingerprints from where...

JASON: Yes. Oh, totally. That thing's been there a long time.

Steve: So, yeah.

JASON: And that's a good point. Virgil in the chatroom says airflow going around the router, I mean, it's keeping it cool entirely.

Steve: Ah, convection cooling for a fanless experience, absolutely.

JASON: Good point. Very good point, Virgil.

Steve: Yeah, I like that.

JASON: Love it.

Steve: Okay. So believe it or not, we had four years ago what was known as the Lenovo Superfish debacle. To remind our listeners what that was, Lenovo began bundling something known as Superfish with some of its computers back in September of 2014. That didn't last long because by the end of February of 2015, the U.S. Department of Homeland Security was advising its removal along with its associated root certificate. So Lenovo came under fire, I mean, immediately upon the discovery of this, for preloading this Superfish, that was the name of this advertising system from some bulk advertising people, I think they were maybe Israeli and operating out of San Francisco, if memory serves.

Anyway, this was part of this preinstalled crap that a lot of PCs are bloated down with to varying degrees. It powered something called Visual Discovery. And this Visual Discovery benefit was meant to help shoppers by analyzing the images on the web pages that they were visiting and then presenting similar products offered at lower prices, thus "helping users search for images without knowing exactly what an item is called or how to describe it in a typical text-based search engine." So this was going to be just a breakthrough.

But to do this, the adware, which was installed locally on the person's PC, on their Lenovo laptop, needed to intercept, decrypt, and inspect all web browser connections, even HTTPS/TLS-protected communications. And to do that it installed its own self-signed root certificate to allow it to impersonate other websites to the PC user's browser. In other words, it inserted itself as a man in the middle. And in order to do that, even with secure connections, and to allow them to remain secure, it needs to be able to produce certificates on the fly for the remote websites you're visiting so that your browser thinks, oh, I'm at Amazon.com, https://www.amazon.com.

So that meant that the certificate that this Visual Discovery, the Superfish stuff, created had to be signed by a trusted cert. Well, it's not a CA, and what it's doing is bad. So it installed its own essentially CA certificate so that it could sign any certificates that it needed to create on the fly. All of this is bad. So it's true that our machines already have a bunch of self-signed root certificates. That's what root certificates are. They are signed by the certificate authorities so that they sort of trust themselves. And that's what trusted CA certs are. But that's why what's in our root store is really crucial.

I mean, unfortunately, we could argue it's not well enough protected. And actually, in some of the commentary and dialogue on the 'Net in the wake of this new event that I'll explain about in a minute, people have been talking about the fact that this should not be so easy for stuff to be installed on the fly. In this case it was preinstalled in the machines that people received. So you say, okay, well, fine. But we'll be talking about something that does it on the fly. So we already have root certificates in our machines. What makes that system safe is that the server which is asserting its identity with a Certificate Authority signed cert, is able to keep the certificate's associated private key safe, since it's safely locked up at the far end of the connection. We have no access to the Amazon.com servers. That's where their private key is that makes all this go. We get the certificate from it, but we don't get the private key.

And even those annoying, and we've talked about these often, these TLS-intercepting middleboxes which enterprises use to peer into all of their employees' Intranet traffic, they're at least self-contained and located in a secured environment somewhere. What made the Superfish transgression so bad was that the equivalent of that server software which is, in the case of Amazon, distantly located, and we have no access to it, it was right there in the same PC because it had to perform the equivalent of being the server for that domain in order to do its man-in-the-middle interception. And that meant that the certificate's matching private key had to also be right there in the PC. And since the private key needed to be used for the software to operate, the private key's own usage key, that is, to decrypt it also had to be present.

So back then it wasn't very surprising when Robert Graham, who's the CEO of Errata Security, he announced that he'd been able to crack the private key for the Superfish certificate, thus effectively breaking the HTTPS security for all affected Lenovo laptops. What this meant was that, even remotely, any site could then be spoofed for anyone using one of these Lenovo laptops. So thus a lot of people were upset. And in fact, it turns out that in the news just recently we've learned that these slowly turning wheels of justice have finally clicked over, and Lenovo has just now agreed to a settlement in a 32-state class-action lawsuit. A federal court has approved a large payout fund for Lenovo, and Lenovo will be required to create a $7.3 million reservoir which is set aside to settle

this class-action lawsuit over exactly this, that they installed this stuff without users' explicit knowledge or permission, and it was a bad thing to do. Okay. So that was then.

Okay. Believe it or not, this has been repeated by something you just wouldn't expect. To give you a taste for it, Ars Technica's headline reads: "Sennheiser discloses monumental blunder that cripples HTTPS on PCs and Macs." Bleeping Computer titled their coverage: "Sennheiser headset software could allow man-in-the-middle SSL attacks." So believe it or not, a headset. The same mistake has just been made, not in trade for some grand sweeping advertising scheme, as was in the case of this Visual Discovery, but only so that the Sennheiser headset software could run in the user's web browser and securely connect to their own local software also running in the same machine.

In other words, with web browsers becoming increasingly fanatical about HTTPS in preference to HTTP, even when the connection is to the same PC made through the unroutable localhost IP, which our listeners know as 127.0.0.1, and actually the whole 127 block, Sennheiser decided that the easiest way to skin that cat would be to run their own secure local web server on the user's machine in order to get like a padlock, to get https://127.0.0.1. And in fact that's the common name on the cert that they created is the localhost IP. And they created the same error.

I mean, there's only one way to do this. If you're going to do that - oh, the other thing that Superfish did I skipped over, I forgot to mention, is that they didn't create a separate certificate for every PC. They could have done that, and it would have still been annoying, but it would have been kept local. And that would have meant that somebody could have only spoofed connections for the PC whose specific certificate they had obtained in order to do so. You could argue, well, if they're able to obtain the certificate from the PC, they would be equally able to plant their own spoofed root CA in the PC, and thus perform the same sort of hack.

So there was really - that wouldn't give them much leverage. The leverage came from the fact that all Lenovo PCs shared the same certificate, thus getting it from any Lenovo PC meant you could spoof for all Lenovo PCs. Sennheiser made the same mistake. Rather than minting, as they could, a unique certificate for every machine, their installer just has one. And even worse, after you uninstall the Sennheiser stuff, it leaves it behind.

So anyway, so the same thing has happened. The guys who discovered this, Secorvo Security Consulting, ginned up a custom certificate of their own. In their report about this they wrote, and I skipped the preamble, they said: "Then we created a new key pair and used our fraudulent CA to issue a wildcard TLS server certificate for hosts in the domains of Google, Sennheiser, and some of Sennheiser's competitors," which allowed them to remotely impersonate any Google or other domains for anyone who had ever installed that Sennheiser software, even after uninstalling it, because it left it behind.

So this was all disclosed responsibly. Secorvo reached out to Sennheiser, said this is not good. So there are updates for the Sennheiser software now. Microsoft felt that this was enough of an issue that they waded in with their own remediation of this. They have an advisory 180029 titled "Inadvertently Disclosed Digital Certificates Could Allow Spoofing." And Microsoft wrote: "Microsoft is publishing this advisory to notify customers of two inadvertently disclosed digital certificates that could be used to spoof content and to provide an update to the Certificate Trust List (CTL) to remove user-mode trust for the certificates. The disclosed root certificates were unrestricted and could be used to issue additional certificates for uses such as code signing and server authentication. More details are here." And then they list a couple links.

"The certificates were inadvertently disclosed by the Sennheiser HeadSetup and HeadSetup Pro software. Consumers who installed this software," writes Microsoft, "may be vulnerable and should visit HeadSetup Update for an updated version of the

HeadSetup and HeadSetup Pro software. As a precaution, Microsoft has updated the Certificate Trust List to remove user-mode trust for these certificates. Customers who have not installed Sennheiser HeadSetup software have no action to take to be protected. Customers who have installed Sennheiser HeadSetup software should update their software via the links above."

Now, this brought me back to something that we talked about. I mean, we've talked about this before, the sort of the background danger of malicious certificates creeping into our own PC's root store because the whole bargain here is that we trust the certificates in our root store because the owners of the matching private keys for those certificates are being trusted with their ability to create certificates for any domain. I mean, this is the big danger, this is the Achilles heel of the whole system is that it's an any-to-one mapping. Any single CA that we trust has the ability to create a certificate for any domain.

There was a tool, RCC, which we talked about before. I went looking around, and the domain that was hosting it is gone now. And it's for sale if anyone cares, but I wouldn't bother. There's a better solution from our friends. Mark Russinovich and Sysinternals have a tool called Sigcheck. And I've got the links in the show notes to the Microsoft page of documentation and also for the zip file containing it. And it's, as always, as are all these little tools, two tiny EXEs, one for 32-bit, one for 64. It's a command line utility, and you give the command "Sigcheck -tv," which is kind of easy to remember. And what it does is it dynamically pulls the most recent CTL, the Certificate Trust List, from Microsoft and then cross-checks it against the root certs you currently have in your trust store.

I, of course, ran it to see, and I was initially surprised by four things that popped up. And then I realized, oh, that's okay because I, for the work that I'm doing with SQRL, I created certificates for my own localhost and my own machine name so that I am able to create HTTPS connections to myself, which is fine. They can't be used for anything else. But what was cool was that this thing found them and sort of reminded me that I had them there. And so I would, just sort of as an audit, as something to do here as we close a busy security year 2018, might be worth - I'm sure you can just google S-I-G-C-H-E-C-K, Sigcheck, and grab it and open a command line and run it. And just make sure that, if it shows you anything, it should be a null list. It should just say no, nothing.

If anything pops up, take a look at what that is and make sure you know why, as I just did, because what it means is that that thing has the ability to, you know, the owner of the matching private key to the certificate that it shows can synthesize certificates that your browsers will trust. And just before I leave I should mention, though, that remember that this won't work on the Chrome browser for Google's own properties. The Google Chrome browser knows, it has the fingerprints of every correctly synthesized Google certificate. So this spoofing of certs never works on Chrome for Google properties. It works for other properties, but just not - you know.

So Secorvo Security Consulting said that they created a wildcard TLS certificate for hosts in the domains of Google, and it's like, yes. And so you could do mail.google.com, but don't try going there with a Chrome browser or all kinds of alarms go off, and that's not good. So anyway, it's interesting how these mistakes keep getting made. I would argue a headset installer has no business mucking with anybody's root certificate store. I mean, it's just ridiculous. There's no need. First of all, there's this confusion about localhost. It's not actually networking. It's using the packet-pushing, IP-addressing, networking architecture sort of to create interprocess communications.

That's all it is when it's staying within your local machine. If you connect with localhost, nothing goes out on any wires anywhere. There's nothing happening external. So all it is, is interprocess communication. Unix has used this with Unix sockets forever. So this

notion that they needed a secure connection to the same machine is nonsense. I mean, it's crazy. And because it's often done without HTTPS, browsers are honoring the ability to not have HTTPS for localhost connections. So it was unnecessary in the first place.

And if they for some reason really, really, really felt that they had to do it, then the installer should have made a certificate on the fly so that the same mistake made by Superfish wasn't being made again, and at least the danger would have been contained to somebody who had planted a certificate in a machine or somehow got the private key for the one that had been minted. The damage could have been constrained. And who knows? Maybe this is another class-action lawsuit. And if a few of them happen, maybe people will learn that this is just not something that can be done. And really it ought to not be possible for an installer to surreptitiously drop stuff into our root store. All kinds of warnings should have been produced. And so that's on Microsoft. This ought to be closed down.

JASON: No kidding. And just to be clear, that Sigcheck, that is not necessarily - so you're encouraging anyone to use this, whether they've installed or used these Sennheiser products or not. This is broader than that. This is checking everything including.

Steve: Yes, yes. Yes, because this story is, of course, about Sennheiser. Superfish was about Superfish. But other stuff, I mean, the fact that we've just learned as an example that installing Sennheiser headset software can plant a bogus root cert in your trust store suggests, ooh, what else might that hasn't hit the news that might be in your machine. And malware could do this, as well. And in fact I think there have been, not malware, but there have been some AV tools which have done this for the same purpose because they want to go in and intercept communications. So it's just it makes sense to audit the root store, our PC's root store periodically just to make sure nothing has crept in when we weren't looking. And Sigcheck, I would argue, would be the official way to do this. RCC was good, but it looks like it's just been abandoned. And again, it's from a third party. The domain has expired. So it's like, eh, let's just stick with Mark Russinovich's stuff, which we know we can trust.

So as we talked about at the top of the podcast, Marriott two years ago in 2016 purchased the so-called Starwood Properties, which was a whole bunch of different things. I know that in your conversation with Bruce Schneier on Friday he mentioned that he was a customer of some Starwood assets. That's W Hotels, the St. Regis, Sheraton Hotels & Resorts, Westin Hotels & Resorts, Element Hotels, Aloft Hotels, the Luxury Collection, Tribute Portfolio, Le Meridien Hotels & Resorts, Four Points by Sheraton, Design Hotels, and even the Starwood-branded timeshare properties. So a whole bunch of very well-known hotel chains purchased two years ago, acquired by Marriott.

It turns out that what we are just now learning is that two years before then, something apparently crawled into the Starwood network and set up shop, a classic APT, Advanced Persistent Threat, which had been there. As a consequence of that, Marriott has been forced to disclose that they're still sort of determining the scope of this. And they said that they had not yet finished identifying duplicate information in the database, but believe it contains information on up to approximately 500 million guests who made a reservation at a Starwood property going back to as early as 2014.

They said for approximately 327 million - 327 million - of these guests, the information includes some combination of their name, mailing address, phone number, email address - oh, so physical mailing address, phone number, email address, passport number, Starwood preferred guest account information, their date of birth, their gender, their arrival and departure information, reservation date, and communication preferences. They said for some the information also includes payment card numbers and payment card expiration dates, but the payment card numbers were encrypted using 128-bit AES.

There are two components which are needed to decrypt the payment card numbers; and, they wrote, at this point Marriott has not been able to rule out the possibility that both were also taken. So the fact that the payment card information was encrypted doesn't mean that the bad guys can't decrypt it. So they said for the remaining guests the information was limited to name and sometimes other data such as mailing address, email address, and other information. They reported this incident to law enforcement and continue to support that investigation, which is ongoing, and they've already begun notifying various regulatory authorities, as needed.

So this doesn't rise to, believe it or not, the largest breach ever; however, it's now solidly in number two. The largest breach ever was that massive three billion user breach reported by Yahoo! back in 2016. But Marriott is now number two. And I guess predictably, because Marriott's well known, the breach is so big, there was an immediate response from Congress. Senator Mark Warner said in a statement: "We must pass laws that require data minimization, ensuring companies do not keep sensitive data that they no longer need. And it is past time we enact data security laws that ensure companies account for security costs, rather than making their consumers shoulder the burden and harms resulting from these lapses."

And I have to say, you know, we sort of roll our eyes when Congress gets involved in anything, or the government, or bureaucracy. But this notion of data minimization, I think that makes a lot of sense to me. I've spoken of it before. And this idea that data doesn't self-expire, and as a consequence databases which are never pruned are able to just accrue data without end for no reason except nobody has said we need data minimization. We need data to expire itself when you could argue it no longer has any business value, and after some length of time. Certainly things, for example, credit card information. After its expiration date, it ought to be removed. It's no longer valid. And you could argue, well, it doesn't have to be removed if it's no longer valid. But it just ought to be part of the process.

So what I like about it is that, if there was legislation in place to require data minimization, it would be real protection, unlike other policies which are sort of too soft, because its enforcement would be easily tested. When a breach occurred, and the information that had escaped was revealed, the responsible company can be asked why exactly are you still retaining all of that data from X time ago, from a decade ago or five years ago or two years ago. What is its ongoing business purpose? And so the point is it would expose companies to a clear violation of a law to which they could then be held accountable, which we don't currently have. And that would force it to be enacted, and consumers would clearly benefit from that happening. So I just think that that part of it makes sense.

JASON: I think that makes sense, although the cynical part of me is like, yeah, but then the companies are going to come back, and they're going to figure out a reason why they keep, you know what I mean, they're going to make it part of their business to keep that longer and to justify it. And I guess that's the point; I guess that would be the point. But once businesses have this data, it's like next to impossible for them to be okay not having it anymore. They would resist very hard.

Steve: I know. They want it. They want it. In fact, we'll be talking about one here in a minute, a company that's called Data & Leads, as in like sales leads. And their business model is aggregating all of this. So, yeah, they don't want to let go of any of it, ever. Maybe it'll be useful. And of course in their brochure they're able to brag about how many people they have in their database. And I think it was 57 million. And so, yeah, they don't want to have those migrate away.

But before we get to that we have, as you mentioned also, this Quora breach. Quora, probably everybody knows, they're the Internet's leading question-and-answer site. They

just announced a problem. Adam D'Angelo posted on the Quora blog with the heading: "Quora Security Update." He said: "We recently discovered that some user data was compromised as a result of unauthorized access to one of our systems by a malicious third party. We are working rapidly to investigate the situation further and take appropriate steps to prevent such incidents in the future. We also want to be as transparent as possible without compromising our security systems or the steps we're taking; and, in this post, we'll share what happened, what information was involved, what we're doing, and what you can do."

And he finished his little introduction saying: "We're very sorry for any concern or inconvenience this may cause." Then, under "What happened," he said: "On Friday" - and that's only four days, four or five days ago. "On Friday we discovered that some user data was compromised by a third party who gained unauthorized access to one of our systems. We're still investigating the precise causes; and, in addition to the work being conducted by our security teams, we have retained a leading digital forensics and security firm to assist. We've also notified law enforcement officials. While the investigation is still ongoing, we have already taken steps to contain the incident, and our efforts to protect our users and prevent this type of incident from happening in the future are our top priority as a company."

Under "What information was involved," they said: "For approximately 100 million Quora users, the following information may have been compromised: account information, meaning name, email address, encrypted hashed password, data imported from linked networks when authorized by users; public content and actions, in other words, questions, answers, comments, up votes; non-public content and actions, for example, answer requests, down votes, direct messages." And he wrote: "Note that a low percentage of Quora users have sent or received such messages."

He said: "Questions and answers that were written anonymously are not affected by this breach as we do not store the identities of people who post anonymous content." He said: "The overwhelming majority of the content accessed was already public on Quora, but the compromise of account and other private information is serious." Under "What we're doing: While our investigation continues, we're taking additional steps to improve our security. We're in the process of notifying users whose data has been compromised. Out of an abundance of caution, we're logging out all Quora users who may have been affected; and, if they use a password as their authentication method, we are invalidating their passwords." So a forced password reset.

"We believe we've identified the root cause and taken steps to address the issue, although our investigation is ongoing, and we'll continue to make security improvements. We will continue to work both internally and with our outside experts to gain a full understanding of what happened and take any further action as needed."

And then, under "What you can do," he said: "We've included more detailed information about more specific questions you may have in our help center, which you can find here." And there's a link that he provided. I have it in the show notes if anyone is an active Quora member. You could find the link in the show notes. Or actually I know that you have probably received email from them because, Jason, you and I got a forward from someone who was affected. And so Quora has been proactive. Actually in this he didn't say that they are proactively notifying everybody who may have been affected, but obviously they are. And, finally, they said: "While the passwords were encrypted," he said, "hashed with a salt that varies for each user" - so per-user salt - "it is generally a best practice not to reuse the same password across multiple services, and we recommend that people change their passwords if they're doing so."

So my takeaway is here is a textbook case of a company with state-of-the-art policies, doing everything right, taking quick action. We don't know how early somebody got in,

that is, we don't know how quickly they figured out that there was something wrong. But we do know that it's not four years. Well, we don't know. But we do know because - I drew four years from the Marriott instance. In the case of Marriott, we do know that somebody has been in the Starwood stuff from 2014.

So we don't know at this point how soon this happened. But it's certainly the case that immediately upon determining there was a problem, Quora jumped and took every possible appropriate step. And we know that their internal technology policies per user salt on passwords is the best that we know how to do. They didn't mention PBKDF2, if their salting process and their password hashing is also deliberately slowed down to prevent brute force cracking of specific account hashes, but we can assume and hope that that's the case. Given everything else that they've done, I wouldn't be surprised. So I think that's all anybody can do.

And just in the third and final of this lump of news we have the records of at least 57 million records in a massive 73GB data breach. It turns out that Elasticsearch is currently the number one searchable indexed database in use by enterprise. The bad news is it is intended for internal server access, does not have an authentication mechanism by default, and it can be and has been exposed, instances of it have been exposed on the public Internet in the past. The Shodan search engine found at least three IPs where identical Elasticsearch clusters were misconfigured for public access. The first of these IPs was found and indexed by Shodan on November 14th of this year, and that open Elasticsearch instance exposed the personal information of 56,934,021 U.S. citizens with information including their first name, last name, employers, job title, email, their physical address, state, zip, phone, and IP address.

Another index of the same database containing more than 25 million records with more of the so-called "Yellow Pages" details had name, company name details, zip address, the carrier route, longitude and latitude, the census tract, their phone number, web address, email, employees count, revenue numbers, SIC codes; and so that looks like a Yellow Pages business directory of some sort. They said: "While the source of the leak was not immediately identifiable, the structure of the field source in the data fields is similar to those used by a data management company, Data & Leads, Inc. And the people who discovered this were unable to get in touch with Data & Leads Inc.'s representatives. And shortly before the publication of this, the Data & Leads website went offline and is now unavailable."

So the coincidence is suspicious. As of today, the database, this Elasticsearch database is no longer exposed to the public. However, we don't know for how long it had been online before the Shodan crawlers indexed it for the first time on November 14th, and who else may have accessed and acquired the data. So anyway, the people who put this up wrote that: "Our goal is to help protect data on the Internet by identifying data leaks and following responsible disclosure policies. Our mission is to make the cyber world safer by educating businesses and communities worldwide on ethical vulnerability disclosure policy. We regularly publish reports," blah blah blah. So they did the right thing by attempting to get a hold of this, not disclosing that this was public until it was no longer public, even though they were never able to officially determine that it was Data & Leads that were the source of this.

So stepping back from this a little, we have three different instances, all that vary in their details. It looks like Quora's system was designed correctly and responsibly, but the bad guys found a way inside even so. It sounds a little bit as though Marriott made a somewhat diseased or ill-advised property acquisition two years ago, which then came back to bite them. In retrospect, maybe they should have performed a more careful forensic analysis of their Starwood acquisition beforehand. And who knows? Perhaps they did, to some degree, and this problem was missed. But if it was, then this would suggest

that their analysis was not deep enough. Maybe it was just pro forma, just for show, just so that they could say that they had done that.

But the Sony incident demonstrates that an advanced persistent threat can get into a network and hide itself and live there for a long time. And then we have what looks like just irresponsible configuration of an Elasticsearch service. Elasticsearch is in use heavily by enterprises, and they are not exposed by default publicly, or there would be many more such instances. So this was a company that probably had theirs misconfigured, and it's disappeared now. But it really means that, as users of the Internet, as participants online, to varying degrees our information is exposed publicly. And as you said, Jason, it is in databases, and these companies don't want to let go of it. And large sums of money are being leveraged for access to this data, this personal data which, I mean, there are companies that are now selling the fact that they have acquired these databases and are making them available.

So I think what most of us do now is to be aware of the realities of our use of the Internet, that to varying degrees our online lives are subject to exposures. Maybe it makes sense to create silos, individual trust silos where we use different identities, different email addresses so that to some degree, when something is exposed, it's not our entire world that is exposed. And I guess just be cognizant of the fact that what is online, anything placed online is subject to eventual disclosure to varying degrees. Even companies doing their best to keep that information safe can still have breaches.

JASON: I think what I learn or what I realize in all of this, taking a look at all of these incidents of hacking, and the Starwood hotels is just one example, like do I think that consumers are going to see this and say, well, I'm not going to stay there anymore? And I think people are just getting so complacent at this point, you know what I mean?

Steve: Yup.

JASON: Like it feels, and I know you know this better than anybody, it feels like an everyday occurrence at this point. And so as that happens, it just becomes more and more muted. So thankfully, you know, hopefully there are actions to hold their feet to the fire because I'm not sure the consumers are even doing that anymore.

Steve: No, I agree.

JASON: Consumers are like, oh, this is part of technology. I guess we just get used to it. And that's a problem.

Steve: Yeah. Well, and one could argue, too, that big as these numbers are, if the credit card information is old, well, I mean, no one wants their credit card information to get lost, but it happens. For whatever reason, I mean, I was an early adopter of credit card purchasing on the Internet. And I've told this story before. I used to fly to Northern California, I still do, annually to visit my family on the holidays. And back in the day I used a travel agent. And when I would give her a call, she'd say, "So, Steve, same credit card, or did that get compromised?" And as often as not I would say, "Well, Judy, yeah, I got a new card because there were some bogus charges on my last one, and so I've had to change my number."

Happily, that hasn't happened. Well, for one reason I'm using PayPal that masks a lot of my purchases now behind that PayPal API, which is a great benefit. But I think that the point is that, even though that can happen, exactly as you said, consumers are still saying, well, the benefits outweigh the risks. And so I would argue a rational consumer is still going to stay at a Marriott-owned property if the hotel is in the location that they want and is available and for the best price. I mean, irrespective of the security side, that

just doesn't factor in. I doubt that it's going to. And I would argue it probably should not, that even though these things can happen, it's like, yeah, well, okay. But the convenience of, I mean, how can you even book a hotel? I guess you can call them. But they're going to enter it into their computer, and you're going to be in their database anyway. So it's unavoidable.

JASON: Absolutely. It's depressing where we are, but we are where we are. So, Akamai. Haven't thought about Akamai in a while. What's up with Akamai?

Steve: Well, they're keeping an eye on things for us. They named something that they found "EternalSilence" because it combines EternalBlue - which is the NSA-created, I guess you'd call it "attack ware." EternalBlue is this exploit for moving among Windows file and printer sharing Windows networks, which has been weaponized and been wreaking havoc. Anyway, EternalBlue and Silent Cookie. Silent Cookie is the name that an exploit of Universal Plug and Play calls itself because of the entry placed in the NAT routing tables of the routers that it exploits.

So Akamai says that over 45,000 routers have been compromised with this what they call EternalSilence because it's EternalBlue plus Silent Cookie. They detected a malware campaign that alters the configurations on home and small office routers to open connections inward toward internal networks so bad guys can access and infect previously isolated computers. And we've been saying this is going to happen. It's just a matter of time. While bad guys were using UPnP only to bounce traffic off of exposed routers, it was like, okay, well, it's a good thing that that's all they're doing, but it's always been the case that they could decide to get meaner if they wanted to.

And that's now happened. They're doing this using the so-called UPnProxy, U-P-N-P-R-O-X-Y, UPnProxy technique, which we've talked about before. Unfortunately, a surprising, I mean, an amazing number of NAT routers have the UPnP port, port 1900, open to the WAN. I just - I do not see any possible reason for that. It is meant to be an internal protocol on the LAN interface that allows, for example, the Xbox famously to access Universal Plug and Play - and I have to make sure I don't say "Universal Plug and Pray" - in order to allow the Xbox for people who want to do Internet networked gaming to accept unsolicited incoming connections. So an Xbox is able to interface with Universal Plug and Play in order to say, okay, open the following ports through to my internal IP on the Intranet, that is, the internal LAN behind the router. That was its intended purpose.

The moment that it appeared, I raised the alarm that there was no authentication in this protocol. And yes, that was done for ease of use so that it was zero config. Unfortunately, if you also make the mistake of binding this protocol to the WAN interface, then you're exposing a zero config, authentication-free-by-design protocol to the public Internet. And this is the result. So when we first talked about this last April, we saw that hackers were using this technique to convert routers into proxies for bouncing DDoS, spamming, phishing, and other traffic.

But in a report published last Wednesday, Akamai says it's seen a new variation of this UPnProxy where hackers are now leveraging UPnP services to insert different rules into these exposed routers' NAT tables. The rules still function as a proxy, but instead of relaying the traffic back out, they allow an external hacker to connect to the SMB, that's the Server Message Blocks, ports 139 and 445, of devices and computers located behind the router on the internal network. Akamai says that something on the order of 277,000 routers, more than a quarter million, 277,000 routers they have found with vulnerable UPnP services are now exposed online. Of those, 45,113 have been modified in this recent campaign they've uncovered.

Okay. So that says Akamai themselves have been connecting to port 1900 and pulling the NAT routing table out and examining it. They found that one particular hacker or

hacker group has spent weeks creating custom NAT entries across these 45,113 routers named "galleta silenciosa," which they said is Spanish for silent either cookie or cracker, thus the name of this exploit. So what they're doing is, once this hacker is able to create this mapping that allows them to access port 139 and 445, then they're injecting the NSA's EternalBlue weaponized SMB exploit into the networks in order to then expand laterally and find all of the machines that have those ports open and are internally using file and printer sharing on Windows machines. Which then Akamai says they have detected "millions of successful injections" during which these crooks connected through these ports to the devices beyond the router. Akamai said that these devices number somewhere in the 1.7 million range.

What the hackers did, Akamai cannot determine as they don't have visibility inside the networks. That would be, I mean, it's one thing to take a peek at the exposed routing tables in the NAT routers. It would be going too far for them to go inside the network. Though the point is anybody can, if you happen to have one of these routers with an exposed UPnP. This has been going on for a long time, that is, the exposure has been. Now, only recently, hackers have woken up to it and said, oh, let's have some fun doing this.

Anyway, so I guess if there's any good news, it's that this does not appear to be a large-scale, nation state-orchestrated hacking operation with any larger goal in mind. Akamai said that the recent scans suggest that these attackers are opportunistic intruders. The goal isn't a targeted attack. It's an attempt at leveraging off-the-shelf exploits to cast a wide net into a relatively small pond in the hope of scooping up a pool of previously inaccessible devices.

So anyway, they wrote that companies and individuals who don't want to be victims of these and future attacks are advised to either disable UPnP service on their routers - yes, please do - or, if UPnP must be exposed on the WAN side, and again I know of no use case for that, they said at least obtain a well-secured router that doesn't use a vulnerable UPnP implementation. And by all means give it an impossible-to-guess username and password. So anyway, Akamai refers to this particular router hacking campaign as EternalSilence, and I have a feeling we've not seen the last of campaigns of this sort.

JASON: Have we seen the last of PewDiePie? This is the most important question I have to ask you today. I know that the episode theme is on GPU RAM Image Leakage. I really think it should have been on PewDiePie. Just my opinion.

Steve: It was a close call, I'll say that. Because even, I mean, for even I to know that PewDiePie, I mean, I've heard that before. I've heard the phrase. I don't know what one is, but I know that there's something...

JASON: What is a PewDiePie?

Steve: There is something. There is a PewDiePie. That's a something. So imagine this. You're apparently a big fan of the Swedish YouTuber and comedian and videogame commentator whose real name is Felix Kjellberg, known on YouTube as PewDiePie. You learn that his number one position on YouTube by subscriber count is being endangered by T-Series, some apparently lame Indian music record label and film company that simply uploads videos of Bollywood trailers and songs. Both YouTuber channels are right around 72 million subscribers. But at the moment, PewDiePie is narrowing. It's down to only about 300,000, and you want to support PewDiePie.

You decide to take matters into your own hands. You want to send out a message to everyone. But you don't have President's Trump's "the world is ending" universal cell phone presidential SMS blaster code. But you are in possession of some modest hacking skills. And you're apparently not very troubled by questions of ethics, morality, or

legality. How do you proceed? You surf over to Shodan and poke around to find a bunch of something. Turns out you find a lot of printers, publicly exposed printers. Perfect. PewDiePie, here we come. According to the hacker, he found three different vulnerable printing protocols on Shodan - IPP, LPD, and JetDirect - with up to, get this, 800,000 vulnerable printers in total. Jason, what is becoming of this world?

JASON: I don't really know.

Steve: It's just [muttering]. So this guy tweets. And what's his name? Oh, his handle is @HackerGiraffe. So @HackerGiraffe tweets: "I was horrified [uh-huh] to see over 800,000 results show up in total. I was baffled," he tweets, "but determined to try this. So I picked the first 50,000 printers I found running on port 9100" - yup, that's the printer port - "and downloaded the list off Shodan." He says: "I then used PRET [P-R-E-T], the PRinter Exploitation Toolkit, on GitHub, which gives hackers the ability to access files, damage the printer, or access the internal network." Whoo, how nice.

However, @HackerGiraffe said that he only wanted to use the kit to print out messages about PewDiePie to "spread awareness" and make sure that T-Series doesn't overtake PewDiePie as number one because of course. He tweeted: "PRET [PRinter Exploitation Toolkit] had the scariest of features: ability to access files, damage the printer, access the internal network, things that could really cause damage. So I had to do this to at least help organizations and people that can protect themselves." So, oh, I guess it doubled as a public service announcement.

The hacker typed up a bash script which runs an exploit kit against the impacted IP, that is, 9100, with commands to print a message, then quit. He then uploaded the script onto his server and left it running. The printed message said: "PewDiePie is in trouble" - which is not something you see every day - "and he needs your help to defeat T-Series! PewDiePie, the currently most subscribed to channel on YouTube, is at stake of losing his position as the number one position by an Indian company called T-Series, that simply uploads videos of Bollywood trailers and songs." I guess they're popular, though.

The message then urged readers to unsubscribe from T-Series and subscribe to PewDiePie, and concluded the message by telling readers to tell everyone they know. And Jason, you are dutifully showing this message on the video of the podcast at the moment. And it went on to say other things. Many people were surprised by this. In fact, one person tweeted that it appeared on the ticket printer of the police station. So, yes, there are many exposed printers on the Internet. Even police ticket printers are online. So, you know, never to allow a good opportunity to go unexploited, within a day or two, the PewDiePie hack, which did generate lots of attention in the Twitterverse, has apparently spawned a new web service over the weekend. There is now a Printer-Spam-as-a-Service, known as just the unimaginative name "Printer Advertising." We have a picture in the show notes of...

JASON: Guerilla marketing.

Steve: ...printer adver- exactly, guerilla marketing. It says: "Secure your spot in the most viral ad campaign in history. We have the ability to reach every single printer in the world." Well, not mine, and I'm sure not those of most of our listeners. And then they said: "Reservations are limited." Uh-huh. So the good news is, if this actually happens, the end result will be certainly the removal of at least some of Shodan's inventory of apparently as many as 800,000 currently exposed accessible and perhaps vulnerable printers from the public Internet.

Andrew Morris, the founder of GreyNoise Intelligence, detected the message, that is, this printer advertising message, in one of his company's honeypots on Sunday. But the spam campaign pushing this ad to exposed Internet-connected printers has continued

through Monday, through yesterday. All of the printer spam originates from an IP address which is quite well known to those who monitor this sort of Internet background radiation, which our listeners know is the term I coined long ago to describe this junk on the Internet that's just never going to go away. It's just background radiation, stuff scanning for, you know - Code Red and Nimda worms, there are still some instances of them alive and scanning.

The IP address is 194.36.173.50, which is known for generating quite a lot of bad traffic. It's scanning for routers for UPnP services, ColdFusion plugins, exposed LDAP servers, web servers, DNS servers, and Memcached servers. So just sort of a potpourri. Anyway, if you start getting spam on your printer, please take that as a heads-up that somehow your printer is exposed to the Internet, and either you or your IT people should fix that because we have talked about the fact that printers are incredibly complex interpreters. And as we know, interpreters are virtually impossible to secure. And printers have lots of known vulnerabilities. They should not be exposed to the Internet. It'll just be another way into your internal private network. Not good.

So we've talked about ransomware, of course, a lot because - I guess it's maybe been supplanted to some degree by cryptocurrency mining that seems to have caught the attention of people. The problem for a long time was how to get paid, or how the ransomers who had encrypted somebody's computer would get paid. Then of course along came bitcoin and solved that problem. It was like, oh, we'll ask for some bitcoin. So one of the questions has been, is this profitable? Does it pay? One of the problems we've noted has been that, first of all, if anybody's got a current backup, then you're certainly more assured to use your current backup to restore your computer than to pay sketchy ransomers for maybe restoring the contents of your files.

There have been instances where, speaking of police stations, all of a police station have had all their computers encrypted with ransomware, and they have paid to have them decrypted because they had to have access to police records that had to be current and had not yet been backed up. So there are instances where ransoms have been paid. As I mentioned at the top of the show, and we have the details here, this week the Department of Justice unsealed a grand jury indictment against two Iranian hackers who are alleged to be responsible for the SamSam ransomware.

As part of this indictment, for the first time, the U.S. Department of Treasury's Office of Foreign Assets Control, OFAC, also publicly attributed cryptocurrency addresses to individuals who were involved in converting ransomware-generated cryptocurrency payments into fiat currency. The Department of Treasury's announcement stated: "While OFAC routinely provides identifiers for designated persons, today's action marks the first time OFAC is publicly attributing digital currency addresses to designated individuals." So that alone is sort of noteworthy.

In this particular case the cryptocurrency addresses are being attributed to Iran-based individuals named - and I'll just call them Ali and Mohammad because their last names are unpronounceable by me - who the U.S. government states have facilitated the exchange of ransomware payments into Iranian currency. The addresses attributed to these individuals are, and I have them in the show notes, they are now public record, and here's where you want to sit down. They contain a combined total of 5,901 bitcoin.

JASON: Dang.

Steve: Yeah, baby.

JASON: Rolling in it.

Steve: Which puts the value of that cache of bitcoin at over 23 million USD. So I hate to say crypto ransomware pays, but, boy. If all of this money was generated from ransom payments, and it seems likely, I mean, we don't know where, may not be U.S. consumers who have been paying because crypto ransomware is a global thing. But, yikes. These guys have made some money.

What's interesting here is that OFAC has also added these two guys and their bitcoin addresses to the specially designated nationals and blocked persons list known as the SDN, Specially Designated Nationals. Which means that U.S. individuals and companies are legally blocked from doing business or conducting any transactions with these individuals, dot dot dot, or with their bitcoin addresses. These sanctions, that means, could also affect non-U.S. businesses and individuals who conduct transactions with them due to secondary sanctions. But on the primary side, that means that it is now illegal for a U.S. citizen to pay the ransom to these guys if you're encrypted with their SamSam ransomware. Which is sort of an odd twist on all of this. I mean, I don't know what you do. I guess back up before you get hit by this annoying stuff.

JASON: I bet they're wishing they had sold it a year ago.

Steve: Oh, yes. Or even two months ago.

JASON: Twenty-three million now.

Steve: Two months ago, because it's dropped in half.

JASON: Oh, I know.

Steve: You're right. It was north of 20,000, but it sat around 6,500 for a long time before this recent crash in value. So, yeah.

JASON: Totally stable.

Steve: Two iOS apps were caught stealing money from their victims. I just sort of wanted to put this on people's radar. It's already been remediated. Apple was quickly notified. The apps were taken down. But it's just sort of an interesting hack. And it would be nice if Apple were to do something to make this less likely to occur. They were both highly rated. Fake rated, but still they had like 4.3 stars, many five-star reviews, glowing reviews, so that anybody in the iOS App Store looking for a fitness app, these things were titled "Fitness Balance" and "Calories Tracker." You did your due diligence. You looked at the reviews. They looked positive. They were fraudulent reviews; but, still, how could you know?

And so what happened was when you installed the app and sat down to use it, it would ask for your fingerprint to access the stored data in the app in order to use it. If you gave it, there was a brief popup showing an Apple Pay transaction of around $100. It was 139 euros or between 99 and 119 USD, depending upon your currency. And then it would disappear. And if you weren't paying attention, you would get billed $100 is the point, is that these apps were charging its users $100 and nagging them if they closed it and didn't do it and kept trying to do it. And the problem is on the iOS phones that have the Home button, that's the same as the fingerprint. And so you put your fingerprint there, and that authorized payment while the phone is on.

The iPhone X, where the button has been removed, there's an option that's on by default that makes it very clear you're making a payment. The screen darkens. It says, just over to the right where the power button is, "Double-click to pay." And so you'd be going, like, whoa, wait, stop. So Apple has on the iPhone X and subsequent devices that don't have

the button, they have made it more difficult to have an inadvertent payment. But, I mean, and I saw screenshots of the popup. Reddit carried a bunch of conversations about this because this exploded quickly while it was there on the App Store before Apple took it down.

And the good news is, if you have a credit card as opposed to a debit card associated with your Apple Pay, we know that credit cards indemnify their owners against fraudulent payments, and so you could certainly get this charge reversed, as long as you knew that it had happened and you then challenged the charge on your statement when it showed up. And maybe you could go to Apple and say, hey, I just got dinged by a malicious app from your store, and see if they could make you whole again also. But anyway, just worth noting that every so often things do sneak past the Apple Store's scrutiny and their curation of apps. Response is generally quick, but you want to keep an eye on these things.

JASON: All right. Got a few little bits here before we head into the main event.

Steve: Yup. So the good news is Mozilla's experiment with DNS over HTTPS is turning out to be a win. They recently posted, late last week, next steps in DNS over HTTPS testing. They wrote: "Over the past few months, Mozilla has experimented with DNS over HTTPS, also known as DoH. The intention is to fix a part," they write, "of a DNS ecosystem that simply isn't up to the modern, secure standards that every Internet user should expect." They said: "Today we want to let you know about our next test of the feature. Our initial tests of DoH studied the time it takes to get a response from Cloudflare's DoH resolver. The results were very positive. The slowest users show a huge performance improvement." And that's significant. The slowest users, meaning that there were users whose DNS servers were more than average slow, and this fixed them so that they saw a big bump in performance.

They said: "A recent test in our beta channel confirmed that DoH is fast and is not causing problems for our users. However," they wrote, "those tests only measure the DNS operation itself, which isn't the whole story. Content Delivery Networks (CDNs) provide localized DNS responses depending upon where you are in the network, with the goal being to send you to a host which is near you on the network and therefore will give you the best performance. However, because of the way Cloudflare resolves names, this process works less well when you're using DoH with Firefox. The result is that the user might get less well localized results that could result in a slow user experience, even if the resolver itself is accurate and fast.

"This is something we can test. We're going to study the total time it takes to get a response from the resolver and fetch a web page. To do that we're working with Akamai to help us understand more about the performance impact. Firefox users enrolled in the study will automatically fetch data once a day from four test web pages hosted by Akamai, collect information about how long it took to look up DNS, and then send that performance information to Firefox engineers for analysis. These pages aren't the ones that the user would normally retrieve and just contain dummy content.

"A soft rollout to a small portion of users in our release channel in the United States will begin this week and end next week. As before, this study will use Cloudflare's DNS over HTTPS service and will continue to provide in-browser notifications about the experiment so that everyone is fully informed and has a chance to decline participation in this particular experiment. Moving forward, we are working to build a larger ecosystem of trusted DoH providers, and we hope to

be able to experiment with other providers soon. We don't yet have a date for the full release of this feature. We will give you a readout of the result of this test and will let you know our future plans at that time. So stay tuned."

And I think this is great. What this says is that traditional DNS over UDP will probably end up ultimately only being used by our OS stuff, not by our browsers. Browsers can themselves, as all of this demonstrates, choose not to use the underlying OS-provided DNS services, but to do their own. And browsers also arguably represent by far, I mean, I don't know what the number is, probably 99% of all DNS activity because, as we know, when we go to some random page on the Internet, these days pages are composed of crap coming from every direction, all which has a domain name that needs to get looked up, to be turned into an IP address for the browsers to go pick up all the stuff that now composes contemporary web pages. As opposed to the underlying OS that, yeah, it's got a few links to Microsoft or various repositories to check for updates and so forth. But not nearly the crazy spread, the heterogeneous access to domains all over the Internet that our web browsers have.

And this sort of automatically secures - not only provides us privacy protection, but it centralizes all these accesses so that it makes sense to push 99% of all DNS over a one-time established secure tunnel to some DoH provider. So I just think we're seeing what is going to end up being an inevitable migration of browser-based DNS to a secure channel to prevent it from being intercepted and snooped on and to give everyone, especially people who had been suffering with slow DNS from typically their ISPs to give them a big performance boost. So I just say yay.

And an interesting piece of gossip, which is all it is at this point. There's no confirmation. Microsoft hasn't said anything. This comes from Zac Bowden over at Windows Central. He writes that: "Microsoft Edge web browser has seen little success since its debut on Windows 10 in 2015. Built from the ground up with a new rendering engine known as EdgeHTML, Microsoft Edge was designed to be fast, lightweight, and secure. But it launched," he writes, "with a plethora of issues that resulted in users rejecting it early on. Edge has since struggled to gain traction, thanks to its continued instability and lack of mindshare from users and web developers."

Now, I should just say, since I'm not primarily a Windows 10 user, I'm using Windows 7, and as we know I'm a Firefox user, I haven't experienced any of that. When I have used Windows 10, I've used the default Edge. It seems fine to me. But anyway, Zac would know. He says: "Because of this," he writes, "I'm told that Microsoft is throwing in the towel with EdgeHTML and is instead building a new web browser powered by Chromium, which uses a similar rendering engine, first popularized by Google's Chrome browser, known as Blink. Codenamed 'Anaheim,' this new browser for Windows 10 will replace Edge as the default browser on the platform, according to my sources, who wish to remain anonymous."

He says: "It's unknown at this time if Anaheim will use the Edge brand or a new brand, or if the user interface between Edge and Anaheim is different. One thing is for sure, however; EdgeHTML in Windows 10's default browser is dead." Whoa. He says: "Many will be happy to hear that Microsoft is finally adopting a different rendering engine for the default web browser in Windows 10." Although on the other hand we know that most people are using Chrome already, so okay.

He says: "Using Chromium means websites should behave just like they do on Google Chrome in Microsoft's new Anaheim browser, meaning users shouldn't suffer from the same instability and performance issues found in Edge today. This is the first step toward revitalizing Windows 10's built-in web browser for users across PCs and phones. Edge on iOS and Android already uses rendering engines native to those platforms, so not much will change on that front."

Oh, and he said: "In addition, Microsoft's engineers were recently spotted committing code to the Chromium project to help get Google Chrome running on ARM. Perhaps some of that work will translate over to getting Anaheim running on Windows 10 on ARM, as

well." And he concludes, saying: "I expect we'll see Microsoft introduce Anaheim throughout the 19H1 development cycle" - which is the current one - "which insiders are currently testing in the fast ring. This is a big deal for Windows," he writes. "Microsoft's web browser should finally be able to compete alongside Chrome, Opera, and Firefox; and those who are all-in with the Microsoft ecosystem will finally be getting a browser from Microsoft that works well when browsing the web." He says: "There's still lots we don't know about Anaheim. I'm sure we'll hear more about it officially from Microsoft in coming weeks."

So that's really interesting. First of all, I guess I agree that having the behavior integrated seems like a good thing. And of course, as I mentioned before, there is, in my own little camp, there is work being done on a web extension for SQRL which is being developed for Firefox and Chrome. Maybe this would mean it would automatically run under Edge, as well, which would be cool.

Just a quick note that Bing was for a while generating a false positive warning about VLC Player. VideoLAN, the creators of VLC, tweeted: "Supposedly, @bing now consider vlc-3.0.4-win64.exe to be malware, which gives an annoying popup." VideoLAN said in their tweet: "This appeared two days ago, and we have no clue how to fix it yet. We've checked, and the binary has not changed and is still correctly signed. TBC." I guess to be continued. Anyway, that was on November 27th. That has since been fixed. But, you know, this happens to all of us. I think it was my Never10 app was generating a false positive malware warning for a while, and it just sort of happens. As we know, malware is becoming a little more heuristic, a little more guessing in its nature, and so false positives are a possibility.

I have two little items in our Miscellany. First off, from Ian Wills, this is probably the single best abbreviation rename I've ever run across: "GDPR renamed Greatly Disproportionate Privacy Response." So anyway, I got a kick out of that.

And many of our listeners have tweeted the news of this Humble Bundle Cybersecurity book deal. When I checked this morning, the time remaining was counting down from just under six days. So anybody listening to this podcast by next week the same time should be okay. Well, minus a day. I'm not sure. So that would suggest that Monday morning of next week it will have expired.

As Humble Bundle purchasers know, these are amazingly good deals at $1, $8, and $15, depending upon which level you decide to purchase. You get DRM-free multiple format, both PDF, EPUB, and MOBI format book deals. I saw among them "NMAP: Network Exploration and Security Auditing"; "Network Analysis Using Wireshark 2"; "Cryptography in Python"; "Hands-On Penetration Testing on Windows"; "Metasploit Penetration Testing Cookbook"; "Mastering pfSense," which that's my favorite router for people to install on various hardware to create a very secure, highly feature-complete premises router; "Mastering Kali Linux"; "Metasploit for Beginners"; and "Mastering Linux Security and Hardening."

And that's like a third of all the books that are available for a very low price. So just in time for Christmas. And you could buy them and give them on a thumb drive to somebody who is interested in security also because you buy them and you own them. And all of the proceeds are donated to charity. So as we know, Humble Bundle does this from time to time, and this looks like a nice lineup.

JASON: Yeah, I love Humble Bundle. They're good.

Steve: Yeah, yeah. And something has never happened before, believe it or not, in the SpinRite world, which is EE, whose "from" line says he's from AU in SF, by the name of Anthony May, wrote a very nice article on, of all places, Quora. And it's long, so I'm not

going to drag our listeners through it all. But it's at Quora.com, "What Needs Repair on a Computer That is Harder Than You Think." And it starts out talking about hard drives and their bad sectors. And I'll just read a little bit. I'll read the beginning and the end.

He said: "Yes, 'spinning rust' hard drives whose design goes back to the 1950s, still going strong throughout this decade despite the rise of solid state drives, will continue well into the next decade." He said: "To be clear, there's not much real repair goes on with modern computers, or tech in general these days. It's deemed not economically viable to repair compared to the cost of replacement, swapping out some module, card, motherboard, drive, et cetera. But hard drives?" He says: "They're spinning death Frisbees, just waiting to gobble your precious data stored on them. And more goes wrong in them than most people realize, their data kept safe only thanks to mathematics."

He says: "'Bad sectors' is a term you hear occasionally, but it's rare to hear someone who actually knows how to fix them because in reality there's only one way I know that has any likelihood of fixing the problem unless you opt for thousands of dollars at a data recovery specialty business, and it's a commercial software utility" - uh-huh, guess what - "which automatically makes people dubious. 'How could software fix hardware?' they scoff." And I'll stop reading at that point. I did encounter the phrase which people have heard me utter before: "Remember, a hard drive doesn't know there's a problem with a sector of data until it tries to read it and discovers that the math doesn't add up anymore."

And then he finishes this posting - I've skipped a bunch of other stuff which listeners may be curious to read. He finishes, saying: "Again respecting the analog-y nature of the magnetic alignment of ferrous particles on the surface of the hard drive and that they can weaken over time, you can exercise the physical hard drive medium by reading the data from the sector, inverting its ones and zeroes and then rewriting that data back to the sector; then reading that data back, reinverting its ones and zeroes, and finally writing that data back to the sector. The net result is the data is exactly the same, but you've pushed every bit through a write of a one and then a zero and then a one or vice versa, leaving the sector freshly written. But at each step the hard drive is monitoring the error-detection math to see if there's any sign of surface defect.

"This is what SpinRite does, the commercial software I mentioned earlier. No, I'm not affiliated with," and he says, "Gibson Research Corporation and Steve Gibson's SpinRite software. But I've used it for nearly 30 years, as have countless other computer pros, and it's rescued countless amounts of data from presumed death because of this analog-y nature of spinning disk hard drives. And," he says, "incredibly" - and I agree - "the same results are now being achieved for SSDs, even though their failure mechanisms are entirely different. But the forward error correction math is still there. The spare sectors are still there. And so corrupted data can still be recovered from both spinning rust hard drives and modern solid state drives." So anyway, very cool posting, Anthony. Thank you for sharing it and letting me share it with our listeners.

Two quick bits of closing-the-loop feedback with our listeners. I'll try to pronounce his name, looks like Frode Burdal Klevstul. He said: "@SGgrc Is there anything in the SQRL protocol that makes CAPTCHAs obsolete?" And, oh, boy, do I wish. But no. It's something that we had discussed over in the newsgroup where we've been hashing this out and nailing down all the details for quite a while, as our listeners here know. There wasn't anything obvious that we could do. And as I mentioned last week, the decision was made to, as much as possible, with only a couple exceptions, keep SQRL minimized to authentication, not burden it with a bunch of superfluous non-auth-related features.

And separating a human from a bot is just not authentication related. It's human body related. I mean, it's are you human as opposed to which human are you, or actually which bot are you because bots could use SQRL also, as we have also talked about in

recent weeks. So anyway, nope, unfortunately, that we decided was outside of SQRL's purview.

And then @themainapp tweeted @SGgrc - and I love this, this is probably my favorite tweet of all time - and @GibsonResearch. He said: "Just finished listening to 'password immortal' podcast. Enjoyed your rebuttal of the paper, but I think you missed a few of the paper's points. I think SQRL will run into usability issues because it's too easy." Okay, well, let's hope it has that problem. Oh, yeah.

JASON: No kidding. It sounds like a good problem to have.

Steve: Sounds like the right problem. We could always throw in a little wrench there to make it a little less easy to use, if that turned out to be a problem.

JASON: Make that a toggle, though, in the settings. Like do you wish for this to be more difficult to use?

Steve: Are you sure you want to log in this easily? You know, exactly. Are you sure you want to be done so quickly with logging in? Okay.

JASON: There you go.

Steve: Okay. So this research, really interesting. Again, as I said, some guys from South Korea turned over another rock, and what did they find? I've got the link to the entire research paper, if anybody wants to go any further than I'm going to here because here I'm just going to, as I do, just sort of cover the big points and share what they discovered and, again, how clever hackers are. So I titled this, well, and this is the title of their paper: "Stealing Webpages Rendered on Your Browser by Exploiting GPU Vulnerabilities."

The abstract of their paper reads: "Graphics processing units (GPUs) are important components of modern computing devices for not only graphics rendering, but also efficient parallel computations. However, their security problems are ignored despite their importance and popularity. In this paper, we first perform an in-depth security analysis on GPUs to detect security vulnerabilities. We observe that contemporary, widely used GPUs, both NVIDIA's and AMD's, do not initialize newly allocated GPU memory pages which may contain sensitive user data. By exploiting such vulnerabilities, we propose attack methods for revealing a victim program's data kept in GPU memory both during its execution and right after its termination.

"We further show the high applicability of the proposed attacks by applying them to the Chromium and Firefox web browsers which use GPUs for accelerating web page rendering. We detect that both browsers leave rendered web page textures in GPU memory, so that we can infer which web pages a victim user has visited by analyzing the remaining textures. The accuracy of our advanced inference attack that uses both pixel sequence matching and RGB histogram matching is up to" - get this - "95.4%" accurate.

They said, okay, in their introduction: "This work considers how attackers can disclose sensitive data kept in graphics processing unit (GPU) memory. We aim to obtain rendered web page textures to uncover web pages a victim user has visited. We successfully reveal such data from modern GPUs NVIDIA and AMD when we enable GPU-accelerated web page rendering in recent web browsers Chromium and Firefox." They said: "For example" - and I have a picture in the show notes, and they refer to that here - "Figure 1 shows the Google logo image of Google.com and a partial dump of rendered web page textures extracted from an NVIDIA GPU used by the Chromium web browser." And it kind of makes it clear that there's a relationship between Google's logo and the debris left behind.

"Although the GPU has rearranged the textures according to its undocumented hardware characteristics, we can infer that the dump originated from the web page because their color patterns are similar. Especially, our combined matching attack can successfully infer up to 95.4% of randomly visited 100 front pages of Alexa Top 1000 websites when a victim uses the Chromium web browser with an NVIDIA GPU."

They explain a little further: "GPUs are important and powerful components of contemporary computing devices. Personal computing devices including desktops, laptops, and smartphones use GPUs for supporting various graphics applications. Graphical user interface, multimedia players, and video games all use them. Large-scale computing devices including workstations, servers, and clusters also use GPUs for energy-efficient massive parallel computations. GPUs utilize a large number of processing cores and a large amount of independent memory for efficiently processing graphics operations and computational workloads. For example, an NVIDIA Kepler GPU can have up to 2,880 cores and 6GB of memory, and its floating-point operation performance is nine times better than that of recent CPUs.

"Programmers can use two types of application programming interfaces (APIs) to access GPUs: the graphics APIs (DirectX and OpenGL) and the computing APIs (CUDA and OpenCL). First, the graphics APIs provide functions for graphics operations such as projection, shading, and texture mapping. Second, the computing APIs provide functions for non-graphics applications such as financial, medical, or weather data analyses; database query optimizations; packet routing; intrusion detection systems; and cryptographic engines.

"The most significant differences between the graphics APIs and the computing APIs are sharing and memory manageability. The computing APIs allow different users to share the same GPU, whereas the graphics APIs only support a single user. A number of users can share the same GPU using the computing APIs in a time-sharing fashion, as the computing APIs demand no dedicated screens, and current GPUs only support sequential execution of different GPU processes. Although some techniques, like VirtualGL, allow remote users to share the same GPU when using the graphics APIs, they warn users of potential security problems, for example, logging keystrokes and reading back images through an X server.

"Second, while GPU drivers manage GPU memory with the graphics APIs, programmers can manually manage GPU memory with the computing APIs, including allocations, CPU-GPU data transfers, and deallocations." In other words, the computing API gives programmers direct access to raw GPU memory. "In contrast," they write, "the graphics APIs provide no functions to manage such memory while providing a set of optimized functions to perform memory-efficient graphics operations."

And they conclude, finally, or they said: "Unfortunately, the sharing and high memory manageability of the computing APIs may incur critical security threats because GPUs do not initialize newly allocated memory buffers. Although numerous studies consider such an uninitialized memory problem in operating systems, no study deals with the uninitialized GPU memory problem. If similar security threats exist with the computing APIs, the threats have much larger impact as multiple users may share the same GPU."

And they explain three points: "In this paper we first perform an in-depth security analysis on GPUs regarding their architectures and computing APIs to reveal any potential security threats. We identify that the computing APIs have a serious uninitialized memory problem because they do not clear newly allocated memory pages, have memory types that programmers cannot delete, and have in-core memory without security mechanisms.

"Second, we develop effective security attacks on GPUs applicable to the most widely used GPUs by NVIDIA and AMD. By exploiting the revealed security threats, our attacks can disclose sensitive data kept in GPU memory of a victim program both during its execution and after its termination."

And they said, finally: "Third, we demonstrate the high applicability of our attacks by inferring browser history of the two most widely used web browsers, the Chromium and Firefox web browsers. Both browsers support GPU-accelerated web page rendering acceleration, which uploads web page textures to GPU memory to increase rendering speed. Our attacks can extract rearranged web page textures of both browsers from NVIDIA and AMD GPUs."

So very much like the muck that we've been mired in all of this year, all of 2018, the very first podcast of this year introduced Spectre and Meltdown. Actually, I think Meltdown was first, and then Spectre soon followed. And we've been dealing with the consequences of subtle flaws in the fundamental engineering of our processors for the sake of their performance all year. Here we have what is essentially an off-chip large memory where an API has been created, the computing API, to leverage the GPU's computational capability when it's not being used for graphics intensive applications. And no thought has been given to the fact that there could be privacy sensitive information. And who knows about, I mean, who knows like if anybody is using the GPUs for crypto acceleration.

One hopes that they are proactively wiping that memory of any crypto keys or any intermediate results from that computation memory before they release it back to the GPU. Otherwise, there's a very good chance that is not just image textures from web browsers, but it's potentially serious information being left over from anybody else who has not proactively wiped their own memory before releasing it back to the OS. So this may not be the last we hear of GPU attacks, thanks to the computing API. Who knows?

JASON: Interesting stuff, Steve. So if somebody wants - is there is anything that you know of as far as automating this? Like clearing it out in an automatic sort of way? I mean, if it's just filling up all the time, automating it seems to be a great approach for the majority of people, anyways.

Steve: Yeah. Now, so if it were deemed important, an update to the API could cause an allocation to be cleared prior to releasing a handle to it for the computation API. What Windows does, and what it sounds like the GPU should start doing, after all, it's got all this computation capability, Windows has a background process which is zeroing unallocated memory in the background all the time. So it's possible to make an allocation of memory and ask for it to be zeroed by Windows. I do. I use it exclusively in the code that I write because you're able to then assume that this will have zeroes and not spend time yourself zeroing it. And it takes no time because Windows typically has lots of memory that it's not actively using so that it just leaves it zero. It goes through and wipes it to zero and then moves it to a pre-zeroed queue where it's allocated from.

That's not currently being done in our GPUs, but this kind of research is what would drive NVIDIA and AMD to say, you know, it wouldn't hurt to have a background thread just going along, zeroing out memory. And if it can, when a computation API call requests memory, give it zeroed memory if it doesn't care, rather than non-zeroed memory. And that could potentially shut this down immediately. So that would be a good strategy.

JASON: Well, there you go. And, boy, you're saying that wouldn't really impact the performance along the way.

Steve: No.

JASON: Because it's happening as-is. It's not like some bulk process that happens.

Steve: Yup, should be completely in the background, yup.

JASON: There you go. Awesome. I think we did it, Steve. We made it.

Steve: Yup. Once again.

JASON: Really appreciate you letting me come on the show with you, man. It's always a lot of fun, and I always learn a ton.

Steve: Oh, Jason, appreciate having you on with us.

JASON: So you can go to GRC.com for all sorts of Steve's amazingness - SpinRite, of course, the best hard drive recovery and maintenance tool that you can get your hands on. Get a copy there. Information, of course, on SQRL, which you talked about a little bit on this show. Details can be found there. Also audio and video of this show can be found at GRC.com. And then, yes, transcripts, which you can't find that on the TWiT site. So go to GRC.com, and you'll find transcripts of this show, and I imagine all previous episodes, all those transcripts, if people want to...

Steve: Yup, they're all there. All 691 previous episodes.

JASON: So people can print them out onto paper and read them on the train as if it was a book.

Steve: That's right.

JASON: So do that. GRC.com. Our website, of course, is TWiT.tv/sn. That's the Security Now! page that houses all the episodes here, audio and video, published on that page, so you could check that out. Record this show live every Tuesday starting at 1:30 p.m. Pacific, 4:30 p.m. Eastern. And I know that time zones have changed since the last time I did this. It's probably not 20:30 UTC anymore, is it. Or maybe it still is. I'm not really quite sure. But look it up. Go to WorldTimeBuddy.com, I think is the site that I usually use, and look up 1:30 p.m. Pacific, and find out when the show is for you. Thank you, Steve, once again. Really appreciate it, man.

Steve: Thanks, Jason.

JASON: We will talk to you soon. We'll see you all next week on another episode of Security Now!. Bye, everybody.

Steve: Bye.