A Decade of Hacks

Description: This week we stumble into Microsoft's own confusion about whether or not Microsoft's Security Essentials will continue receiving updates after January 14th. We look briefly at the year when ransomware happened. We revisit the Avast and AVG Mozilla extensions to see how they're doing. We look at the just-announced big news for Apple's and Google's bug bounty programs for 2020, and also at Mozilla's addition of another very appealing DoH provider (which Leo apparently likes). We provide a nudge to Drupal site masters to update their Drupal Cores RIGHT NOW. And then we conclude by revisiting this past decade - spanning 2010 to 2019 - and the many hacks we've explored during these previous 10 years.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-746.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-746-lq.mp3

SHOW TEASE: It's time for Security Now!, our last show of the year, but also our last show of the decade. Steve's got security news and then a look back at what a crazy decade it has been, next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 746, recorded Monday, December 23rd, 2019: A Decade of Hacks.

It's time for Security Now!, the show where we can - whoo, I'm in a New Year's mood, I can tell you that - the show where we cover your security and privacy online with this guy right here, Steve Gibson. In years past he's been known to dance with Captain Kirk on New Year's, but that won't be happening this year.

Steve Gibson: Without any alcohol, either. I don't know what the heck was going on.

Leo: That was awesome. That was awesome. That was a New Year's Eve marathon episode that we did a few years ago. Hello, Steve. Merry Christmas. Happy New Year.

Steve: Ho ho ho, Leo. I think you're happy because we're all getting a week off.

Leo: Yeah.

Steve: While we're celebrating our holidays. And we're going to have "best ofs." But we're going to end this decade, this being the last podcast of the decade, with a retrospective focus on - wow, it was really fun for me to put this together. I used some research that the ZDNet folks had done, and then of course added a lot more of the techie details and took out some things that were unnecessary to say, but pulled together a really nice sort of decade of hacks, which is today's topic. Although we do have a bunch of news of the week, which I'm going to spend less time on than I normally do because there was just so much that happened in the last 10 years. I mean, and so many of these you're going to, I mean, they'll just put a big smile on your face when we run across like Diginotar and things like that.

Leo: Oh, yeah. Oh, yeah. The Hong Kong Post Office will probably figure in this, yes.

Steve: So we have a great podcast for our listeners. And this is also a commercial-free podcast, so we will make use of the time that is saved. And, oh, such a fun Picture of the Week that we will get to. So we're going to talk about stumbling along with Microsoft over their own confusion about whether or not Microsoft Security Essentials will continue receiving updates after January 14th. We're going to look briefly at the year when ransomware happened. And actually we come back to that at the end of this podcast because of course that was this year. We revisit Avast and AVG's Mozilla extensions to see how they're doing, which remember when they got yanked a week or two ago. We look at the just-announced big news for Apple's and Google's bug bounty programs for 2020; also at Mozilla's addition of a very appealing DoH provider, which apparently you like, Leo, because they've got your endorsement on their home page.

Leo: Oh, boy. I hope it's somebody I like.

Steve: We provide a nudge to Drupal site masters to update their Drupal cores right now. And we'll then conclude with revisiting the past decade of all the things that we've talked about on this podcast, which is pretty much everything that happened in the past 10 years.

Leo: Holy moly. I don't know how you're going to fit all that in.

Steve: I have no idea.

Leo: Well, it's good, because this show has to serve for a couple of weeks.

Steve: Yes. Our listeners could take it in two pieces, if they wanted to, knowing that there will be a "best of" episode as the official Security Now! podcast next week.

Leo: Yes.

Steve: But when I happened to be on the SQRL forums, I saw a posting by Paul Holder, who is one of the team of moderators that I have - the SQRL forums are moderated in order to keep all the crap off of them. It just is necessary to do that these days. And so I've got - I asked from the SQRL newsgroup, as I was setting up the SQRL web forums,

for a show of hands, who wants to be a moderator. And I got about 15 or 16 people who'd been long-term contributors to the SQRL effort, so I sort of knew them by their participation. And so we've got a group of people who are pretty much always there, which is nice because it means that people who are posting don't have to wait long for someone, a moderator to come along and say, oh, yeah, that one's fine, and to let them in.

Anyway, Paul is one of those people. And his posting on Thursday said: "I just converted my TWiT.community account to SQRL login via Jose's OAuth 2 provider." He said: "It's pretty sweet. Time to add it as one of the sites supported on a list somewhere." So actually that spurred me to agree that we need to now have over in the SQRL forums, in GRC's SQRL forums, we need to have a place where people can post SQRL sightings where they've seen things. And of course then I followed the - he posted a link to a thread over in your community, Leo, where you were very active apparently in the past week, bringing SQRL online on your forums.

Leo: Well, and we talked about this last week. As you know, Jose Gomez had written an OAuth plugin for his Discourse forum. We use the Discourse software. It has an OAuth 2 plugin as part of its installation. So it was actually fairly simple. Jose runs a SQRL authorization server. I take it that somebody has to run that. And I signed up with my SQRL account and modified the settings in the TWiT Community, and there you go. You see, you can login with email or login with SQRL. And if you have a SQRL account, which you could set up in a variety of places - are there a lot of OAuth providers for SQRL?

Steve: No. As far as I know, Jose's is it.

Leo: Jose's the first, okay.

Steve: Yes, yes.

Leo: So I'll give you his address, it's sqrloauth.com, sqrloauth.com. It's free to set up an account. Once you set that up, you'll have a SQRL login, and then you can use that to join our TWiT Community forum.

Steve: Very, very cool.

Leo: One disadvantage, it's a little unfortunate, it's not your fault, SQRL's fault, or I don't even think it's Jose's fault. You can't use two-factor if you're using SQRL. I know you don't need it with SQRL. But SQRL does allow you, and SQRL OAuth allows you, to have both, to tie your regular login to your SQRL login. And if you had two-factor set up with your original password-based TWiT Community, which we certainly recommend you do, you have to disable that in order to use SQRL. And so that somewhat reduces your security because there still is that password login, no longer with two-factor.

Steve: Oh, I see what you mean. Yeah, yeah, yeah. Right, right. Because you had to take two-factor off of your password-based login. So I get it. Right, right, right.

Leo: So I don't, I mean, SQRL is effectively as secure as two-factor because of course you have to have somebody's SQRL private key in order to use his SQRL account, and presumably people are pretty careful with distributing that.

Steve: The SQRL private key and the password that you use to authenticate your private key.

Leo: That was one thing that kind of surprised me. Is it always the case that I have to use the first few letters of my SQRL password to authenticate?

Steve: Yeah. The problem is essentially you're giving SQRL the permission to be you on the Internet, to impersonate you. So somebody could walk along, if your laptop was left unattended and go, oh.

Leo: Oh, yeah, use your SQRL, yeah.

Steve: Yeah. Now, in the case of biometrics we don't have that. So if you had a face or a thumbprint or something, then that serves as your "Yes, it's me." But even then, per authentication we just ask you to look at your phone or put your thumb on it because we don't want someone else to come along.

Leo: So it is two-factor. You do actually have two factors.

Steve: Yeah, yeah. There still is that, yes.

Leo: Yeah. So I guess what you'd probably want to do, I don't know how you would do this, is delete your password login to that site and just use your SQRL login.

Steve: Well, and we have anticipated this. There is a setting in all SQRL clients saying, "Please disable all non-SQRL authentication." So in instances where - and, see, this doesn't help you because you've got a problem because you're using OAuth and a SQRL provider. But at some point somebody will implement SQRL natively for Discourse. And when that happens a user can turn that checkbox on, and it solves this problem because we always have this problem of security being about the weakest link.

And so, yeah, it's really great that you've got the security of SQRL. But if you still have username and password, that could still be hacked. So the idea is, after SQRL users become comfortable with SQRL, they can just turn that setting on in their client. And as they visit sites and use SQRL there, the site sees, oh, I've been asked to disable everything but SQRL. And at site's decision, we have no way of forcing sites to do that, but there are a lot of things that I put in there because they're the sort of things where it's impossible to add them later because not everyone will support them. So it's better to have them in at the beginning, even if no one supports them, because then they might. But it's part of the spec. So, yeah, that has been anticipated.

Leo: Nice. Well, now you can use SQRL to log into TWiT.community, I'm happy to say.

Steve: Very cool. Thank you.

Leo: Thank you, Jose. Well, thank you Jose. He's really the one who did all the heavy lifting.

Steve: Yeah. Well, it's going to help - it helps the spread.

Leo: Yup.

Steve: So in apparently a reversal, it now looks like Microsoft's Security Essentials - which we just talked about last week, bizarrely enough, being discontinued, even for enterprises that chose to pay for ongoing Windows 7 security updates. Apparently Microsoft has reversed themselves. I mean, as far as we know. We know this thanks to Computerworld's Woody Leonhard, who posted the question to Microsoft. According to an official post, the company will continue to ship updates to Microsoft Security Essentials after Windows 7 demise. Now, what we still don't know is if that applies to all Windows 7 users or only the extended security update users.

So what Woody explained was, in his Computerworld posting, he said: "Late last week, I talked about a discrepancy in Microsoft's promised handling of Microsoft Security Essentials as Windows 7 reaches end of support. An internally inconsistent official announcement seemed to say that MSE signature file updates would stop, even for those who have paid for extended security updates," he says, "which is absurd. Why would Microsoft stop updating its antivirus program even for people who are paying to continue receiving monthly rollup patches?" he said.

So then last Tuesday Microsoft held an "Ask Me Anything" session, as Woody termed it, for the Win7 forlorn, on the Microsoft Tech Community Forum. Woody asked: "Can you confirm that Microsoft will really, for sure, cut off Microsoft Security Essentials malware signature updates after January 14, even if you're paying for extended support?" Microsoft engineer Mike Cure provided an official response: "MSE will continue to receive signature updates after January 14th." And he cited a Windows 7 Support FAQ which says: "Microsoft Security Essentials (MSE) will continue to receive signature updates after January 14th, 2020. However, the MSE platform will no longer be updated."

And so, again, that makes me now ask, okay, wait a minute. So if the platform isn't updated, will the signature updates go to everyone? Okay. Then also during the same Ask Me Anything, someone, @Brian, responded by referring to the Extended Security Update FAQ which asks the question: "Will Microsoft Security Essentials continue to protect my PC after end of support?" The answer: "No, your Windows 7 PC will not be protected by Microsoft Security Essentials (MSE) after January 14, 2020. This product is unique to Windows 7 and follows the same lifecycle dates for support." So Woody wrote: "That's an obfuscating piece of bafflegab, subject to whimsical..."

Leo: I like that word.

Steve: "Bafflegab." Yeah, I had to do a double-take on that.

Leo: I'm using that from now on.

Steve: I like bafflegab.

Leo: It's bafflegab.

Steve: Bafflegab, embaffling me, it baffles - "...subject to whimsical interpretation, as I described in the Computerworld article last week." That's Woody speaking. Mike Cure then clarified the situation by promising: "I'll get the ESU FAQ corrected as soon as possible." And then Woody concluded by noting that: "As of early last Wednesday morning" - when he had posted - "nothing's been corrected." And he says: "Those of us who actually like and rely on MSE are still hanging on a limb."

So we don't know. I don't think even Microsoft knows. They don't seem to know what the other hand is doing. And of course no one has addressed whether those who don't pay for Security Essentials - so what Microsoft is doing is they're separating the signatures from the platform. They're saying that the platform, the MSE platform itself won't be updated - not that I know that's being updated that often - but that the signatures are separate from the platform. So maybe we'll all keep getting them, or maybe not. I don't know.

So we'll have to wait till next month, and then we'll find out. Actually month after next, I guess, February. It'll be mid-February, the February Patch Tuesday. And also next month is going to have a late Tuesday. Actually we already know that because the first podcast of Security Now! is the 7th, which is the latest it can ever be for the date on a month, which means that of course the 14th is a week later, which is the final, the second Tuesday of January. So it'll be February that we find out whether anybody got updated who still has Windows 7 and isn't paying for the updates.

The second piece I had really should go at the end, the very, very, very end because it's just sort of about - it's an update from Armor, who's been following ransomware stuff, about just what a problem this has been. Eleven more school districts have been hit by ransomware since October something - I have it in the notes here - which is when they updated their cutoffs.

So as a consequence, standing back and looking at 2019 overall, a total of 72 school districts or individual educational institutions have publicly reported being a victim of ransomware which impacted 1,039 individual schools. In the show notes I have a map that's showing a lot, and looks like the upper East was really badly hit. That may have been a consequence of a stronger bias toward managed service providers, MSPs, since we noticed the pattern that anyone who used managed service providers, and we do know several instances where school districts were, that was just like the site of contagion for many of these ransomware attacks. If somebody could get into a managed service provider, they could end up bankrupting the managed service provider because they would lose so many of their clients when all of their clients got zapped as a consequence of all being serviced by a single entity.

And the other thing we saw this year is that the security provisions really were lacking. That is, the MSP gets infected, and then somehow they've got access, like as we have seen, admin rights into the networks of their clients. Which is just a bad idea. If there's a takeaway for our listeners this year, it would be, if your company is using a managed service provider, figure out how to throttle their rights because, unless they really, really need to have admin privilege, you really want to put a throttle on that.

We talked about how Avast and AVG had been removed from Mozilla's store, or Mozilla's add-in repository for Firefox, after Wladimir Palant, the creator of Adblock Plus, took the

time to reverse engineer the backhaul communications of those two plugins and also a couple other, like is this the best price extensions that they offer, and found that they were sending way more data back to the mothership than was accounted for by the service they were offering of letting you know that a URL you were visiting should not be trusted. The extensions are back, and they reportedly behave themselves far better than they had been. We haven't yet heard back from Wladimir. But others have looked and reported that just the URL and a much smaller bit of binary data is being sent back. Still don't know what that is. But it presumably was good enough to satisfy Mozilla.

And as I recall, Mozilla was requiring people to now provide the source code for their add-ins, which would mean that Mozilla would have been able to take a look at what that binary data being sent back actually is. Avast had a standard CYA, you know, our privacy is our number one mission for our listeners, blah blah blah. Okay. Still seems like a bad idea to have somebody, your AV system sending back the URL of everywhere you visit to a provider, especially when we know that they purchased a data broker that is selling in anonymous detailed tracking information.

Basically, all AVG and Avast users are the source of anonymous blow-by-blow tracking of some commercial entity that is offering this for sale. And we know because other providers do this that it's not necessary to take this approach. You don't actually have to send back the click trail of all of your customers. You can do it locally. And as we talked about, Firefox does it, Google does it, where they have a local store of known baddies which they update periodically. And before declaring a page bad, then they do a proactive check just to make sure it's still bad, which seems like a much better thing to do than send back every single click you make, every tab you click on, when the tab loses focus and regains focus. And as we saw, it's just like way too much.

And the revised extension, it puts up a page, I have it here in the show notes, says getting proactive permission, it says - this is Avast online security saying: "Allow us to protect you by scanning web addresses. If you agree, we'll look at the addresses of the websites you visit so we can tell you if those sites are safe. See our Privacy Policy." And a big green, "Yes, Scan Web Addresses." Or, dimmed out, "No Thanks." They're not being really explicit about the fact that it's not your local installation of AVG which is doing the looking, but rather sending back to home base everything you do. So, okay, fine. Anyway, Leo, you and I are just about as down or negative on Avast and AVG as we could be.

Leo: Yeah.

Steve: So enough said.

Leo: You don't need it. You don't want it.

Steve: Yep. As for Google and Apple, who have both done a revamp for 2020, just recently announced their bug bounty awards. On the Google side, last Wednesday Google announced their plans to revamp their six-year-old patch rewards program started in 2013. At the time, Google announced that it would provide financial aid to open source projects if and when and after, significantly, they implemented security features. The project maintainers had to apply, provide a plan for the feature they wanted to implement, and Google would commit to a financial reward after they had actually implemented the features.

But starting the first of the year, Google will be changing how this program works to provide financial support upfront, even before projects implement the security features to which they commit. The rationale behind the change is the recognition that many open source project maintainers prioritize features that will be implemented based upon the sponsorships they receive. Such sponsorship is prevalent, as we know, in the FOSS, the Free Open Source Software community.

So, for example, if a company needed a particular feature added to an open source project, the company usually donates to the project with the condition that the maintainers implement the feature which the company is essentially paying them for with a higher priority before other features. Therefore, by Google providing its funds now upfront, Google is able to provide project maintainers a way to fund their work while prioritizing security features at the same time, rather than relying upon the largesse of wealthy corporate entities whose needs for project features may be more self-serving and less security focused.

So Google will continue, is going to continue to push security, but they're saying we're going to give you money as a sponsor to sponsor the development of the feature ahead of time. Certainly Google can afford this. There are two different fundings that project maintainers can request via this Patch Rewards program. The smaller is $5,000, meant to motivate and reward a project for fixing a small number of security issues like improvements to privilege, separation, or sandboxing; cleanup of integer math operations; or more generally fixing vulnerabilities identified in open source software by other bug bounty programs. So that's $5,000.

The larger, $30,000, meant to incentivize a larger project to invest heavily in its own security, for example, providing support to find additional developers or implement a significant new security feature like full-blown compiler mitigations, that sort of thing. So much more of a salary-ish sort of thing when a much bigger chunk of work wants to be done. And I'm sure that Google will keep an eye on it and verify that it gets done. But they're not requiring it upfront. So that's cool. I have a form for anybody who's interested in applying in the show notes. You fill out the form to apply for the grant. And if Google decides that your project looks like it's worthy, you can get it.

Apple, they've opened their previously closed bug bounty program to all security researchers and have now published their official rules. They talked about this last August, that they were going to be doing that. Well, that has happened. On Friday, Apple formally opened its bug bounty program to all researchers. They had announced it, as I mentioned in August, during Black Hat. There's a big splashy page on the developer site at Apple, headlined "Apple Security Bounty." Up until this time, the bug bounty program was strictly by invitation only, and only accepted iOS security bugs. Now Apple will accept vulnerability reports for a wide array of products, including iPadOS, macOS, tvOS, watchOS, and iCloud. So basically the iOS platform in all of its variations, plus Mac and iCloud.

And the company has increased its maximum bug bounty from $200,000 to, wait for it, $1.5 million. Actually, there's even a way to, I think - oh, no. That would be if you get the extra 50% bump, I'll explain in a second, so depending upon the exploit chain's complexity and severity. However, Apple's not handing out such high rewards casually. The rules are strict, and they've set what those in the industry regard as a high bar for earning the top rewards. On the other hand, it's $1.5 million. So that would be good.

To be eligible for the top prizes and various bonuses, researchers must submit clear reports which include a detailed description of the issues being reported, any prerequisites and steps to get the system into an impactable state, a reasonably reliable exploit for the issue being reported, and enough information for Apple to be able to reasonably reproduce the issue. So of course that's pretty much the Pwn2Own

requirements; right? I mean, that's the set of things that anybody winning Pwn2Own has been able to do in the past. And so now we're talking about some substantial reward rather than just say, hey, look, I got a laptop.

So those bugs which are novel, affect multiple platforms, work on the latest hardware and software, and impact sensitive components are more likely to net the top $1.5 million reward. And vulnerabilities discovered and reported in beta releases will also be highly prized, with Apple willing to add a 50% bonus on top of the regular payout for any bug reported in a beta release. And as we talked about this at the time, because we did get a sense that this was going to happen, this seems entirely rational since it would incentivize researchers not to wait until something is actually out and shipped, but to get onboard and help Apple find these problems before they ever ship them in the first place, which it makes sense that Apple would and should be willing to pay for.

The 50% bonus is for regression bugs where older and previously fixed problems return for an encore, as well as bugs that are found during beta. Also, since takeover bugs requiring no user involvement are the most sought after by the likes of Zerodium, the discovery and reporting of those will also bring researchers top money from Apple. So long as the researcher is able to provide a fully working exploit chain for these types of submissions, they stand to qualify.

Under "Eligibility," Apple says the party must be the first party to report the issue to Apple Product Security, provide a clear report which includes a working exploit, not disclose the issue publicly before Apple releases the security advisory for the report, which of course is generally released along with the updates. Issues that are unknown to Apple and are unique to designated developer betas and public betas, including regressions, can result in that 50% bonus payment.

And so Apple's looking for security issues introduced in certain designated developer beta or public beta releases. And they're saying that not all developer or public betas are eligible for the additional bonus. And then they're very interested in refinding anything that they broke. Regressions of previously resolved issues would also get payment.

So, for example, I have in the show notes - and Leo, you were showing it a second ago - sort of the breakdown, the menu of payment schedule. iCloud: Unauthorized access to iCloud data on Apple servers gets someone who finds a way to do that $100,000. A device attack through physical access, if it's a lockscreen bypass, 100,000. If it also allows user data extraction, that's a quarter million. A device attack via a user-installed app, so if an app can be installed and obtain unauthorized access to sensitive data, $100,000. If the app is installed and obtained kernel code execution, $150,000.

A CPU side-channel attack gets a quarter million. Then we have network attack with user interaction, one-click unauthorized access to sensitive data over the network, $150,000. One-click kernel code execution through a network connection, a quarter million. And the final category, where the big money is, network attack without user interaction, a zero-click access to the kernel with physical proximity, and I don't know what they mean by that, "with physical proximity."

Leo: That would mean like a Bluetooth or WiFi exploit, maybe?

Steve: Yeah, maybe. Network attack without user interaction, zero-click, oh, radio to kernel with physical proximity. So I think you're right, Bluetooth or WiFi, quarter million dollars. Zero-click unauthorized access to sensitive data, half a million dollars. And, finally, zero-click kernel code execution with persistence and kernel protection bypass, that brings you the million dollars. And if to that you found that in a regression - or what

was the other thing? It was - or in a beta. So either you found something that they broke, or you find a zero-click kernel code execution with persistence and kernel protection bypass, that's where you get - and that's in beta software - $1.5 million for the big payout.

So anyway, it's nice that this is happening. As we've said, the problem has been that the commercial entities have not been able to outbid the likes of Zerodium, so a hacker who's trying to make a career from finding security flaws by being an ethical bug hunter - and as we've talked about, it's entirely possible now to do that - they're going to look out for themselves. And so the tendency would be to sell into the gray market like Zerodium rather than help Apple find their own problems. So it's very cool that this is now something that Apple's doing.

And Leo is endorsing a new DoH provider offering.

Leo: This I've got to find out about. It's probably a sponsor.

Steve: If you scroll the show notes down, you'll see your endorsement. Mozilla has expanded its DoH provider offering. Last Tuesday Mozilla announced the addition of a second DoH server provider to its previous list of one, which of course we know was Cloudflare. Users can now choose between Cloudflare and NextDNS.

Leo: Huh. They might be quoting me about DoH as opposed to NextDNS.

Steve: Ah, that's a possibility. Yeah, that would, you're right, that could very well be. So Mozilla said: "Firefox announces new partner in delivering private and secure DNS services to users."

Leo: I guess I should use it and see.

Steve: Actually, it sounds really cool. "NextDNS joins Firefox's Trusted Recursive Resolver (TRR) program committed to Data Retention and Transparency Requirements that respect user privacy. By adding a second provider, NextDNS, a relative newcomer startup founded just this past May, Mozilla has not only added an alternative, but has lifted its promised Trusted Recursive Resolver program off the ground." Mozilla says that its Trusted Recursive Resolver program matters because: "DoH's ability to encrypt DNS data addresses is only half the problem we're trying to solve. The second half is requiring that companies with the ability to see and store your browsing history change their data handling practices." And now of course...

Leo: We're sending them a takedown notice because I've never recommended them. I never even heard of them.

Steve: Ah, okay.

Leo: Although it sounds pretty good; right? It's okay?

Steve: It does sound pretty good.

Leo: Okay.

Steve: Yeah.

Leo: I don't know why they're including me in this. Geez.

Steve: Well, I went there, and I said, oh, there's Leo.

Leo: That's the problem. I mean, you can't just put my name on something.

Steve: Yeah, yeah. Well, and it's a younger Leo. But I think that's a picture of your...

Leo: That's my standard headshot, but I just...

Steve: Floating around the Internet, yeah.

Leo: Yeah, it's probably from - I don't know where they got that.

Steve: Anyway, so as we know, it is likely that ISPs are going to respond to DoH by bringing up their own DoH servers so they can once again capture their customers' DNS. And I think this is why Mozilla is making this push, this notion of a Trusted Recursive Resolver program. Although it doesn't really ring. It doesn't fall off the tip of the tongue very easily. Trusted Recursive Resolver. Besides, most people don't know what a recursive resolver is, let alone why they need to trust one.

But in any event, so the requirements to qualify for Mozilla's Trusted Recursive Resolver program are only collect data, for example IP addresses, for the purposes of running the service, and don't keep it for longer than 24 hours. Publish a privacy policy explaining this. Do not block, modify, or sensor websites unless required to by law. So NextDNS may be apparently not endorsed by Leo. And NextDNS, you're bad people. You should not have done this.

Leo: Uh, yeah. By the way, you pay for this. It's $2 a month.

Steve: Ah, okay. May particularly appeal to more capable tinkerers like Security Now! listeners. So what's the appeal? Control and visibility. Users who sign up for an account are given an inordinate amount of control over what gets blocked and what doesn't, including being able to create domain allow/block lists, and sign up for a range of public advertising and tracking and filtering lists. Techie users can even block specific applications, as well as view their traffic logs, all providing a level of control and visibility into DNS that is very unusual for DNS providers of any type.

So of course the reason, as we know, that so many ISPs are chafing at the pending loss of access to their customers' DNS, is the reason why putting the power of DNS management into the hands of those who want it and know what to do with it makes sense. So anyway...

Leo: They're just going to replace my name with your name now.

Steve: Oh, no. Agh, you're right.

Leo: Well, the quote that they have on the page does not refer to them.

Steve: Right, it's very generic.

Leo: The other two do mention them by name, but I don't think I've ever heard of them.

Steve: Well, and Leo, they put you first. You're on the left.

Leo: Geez.

Steve: Of the three, so...

Leo: Holy cow.

Steve: Yeah.

Leo: Well, maybe if anybody remembers me ever saying anything about them, let me know. Otherwise we're sending them a cease-and-desist.

Steve: Well, and they've not been around. They're only been around since May, so a little over six months.

Leo: But, hey, at least they're a good company.

Steve: They do sound like a good company. And, I mean, the fact that Mozilla is vouching for them, that says a lot.

Leo: Well, are they really?

Steve: Yes. Oh, no, no, they are. I mean, Mozilla has added them officially to their Trusted Recursive Resolver list, meaning that these people have agreed to Mozilla's rather strict requirements over the lack of logging and tracking that this company will do.

Leo: They're not even following me on Twitter. I don't know how they even found me. The founder was a director at Netflix and co-founder of Dailymotion, Olivier Poitrey. It's French. It's interesting. Hmm. I think they're French.

Steve: Well, I imagine your face will be disappearing from their website before long.

Leo: Yeah, the founders of Dailymotion, the two of them have founded it, which is kind of the French YouTube or European YouTube. Huh.

Steve: So Drupal users, the Drupal admins within the sound of my voice need to update their Drupal Core immediately, if they haven't done so in the last week.

Leo: Oh, god.

Steve: The Drupal Core team have released four critical, well, one critical and three moderately critical patches to the core. So that's just a heads-up. The latest release of Drupal 7.69, 8.7.11, or 8.8.1 will prevent remote hackers from compromising web servers. So this is why it's critical. There's a bug in the Archive_Tar library that Drupal Core uses for creating, listing, extracting, and adding files to tar archives. If a site in no way invokes Archive_Tar, then you're okay because that's, for this critical patch, that's where the problem lies. And Drupal Core by default does use this Archive_Tar library. There's a problem in it. So if somebody were to arrange to upload a .tar, a .tar.gz, or .bz2, or .tlz files, that can take over your server remotely. So time to update, if you haven't.

Okay. I blew through the news so that we would have time to play with our decade of hacks. 2010, Leo, the year is 2010, and we had fun with Stuxnet.

Leo: Oh, yeah. And by the way, this show was, what, five years old in 2010, something like that; yeah?

Steve: I think, 10 years ago, no, so nine years ago, yeah, not quite five years. I think it was - was it two years old?

Leo: Oh, okay.

Steve: Because we're in year 12 now.

Leo: We started in 2008? Yeah.

Steve: I think that sounds right.

Leo: That sounds right.

Steve: But still, so it was fresh. It was a fresh podcast.

Leo: Stuxnet, our first really big story.

Steve: It was a big story. Yeah, because really the Honey Monkeys, eh, you know. But, yes, Stuxnet. We are now pretty certain that the Stuxnet worm was co-developed by the U.S. and Israeli intelligence services as their means to sabotage Iran's nuclear weapons program, which had been ramping up in the late 2000s and was, like, beginning to be a worry in 2010. As we covered at the time, Stuxnet was cleverly designed to ride on thumb drives as a means of jumping the air gap to non-networked computers. It was specifically designed to destroy SCADA, that's Supervisory Control And Data Acquisition (SCADA) equipment, and in this case the centrifuges which were being used by the Iranian government for its nuclear fuel enrichment process.

And it did. It successfully destroyed equipment in several locations in Iran. Though there were other cyberattacks carried out by nation-states against one another before 2010, Stuxnet was the first incident that grabbed international headlines and really sort of marked the entry into a new phase of cyber war, moving from simple data theft and information gathering into the actual physical destruction of equipment. So, yikes.

Also 2010, Operation Aurora was the hack that changed Google. Although these attacks by the Chinese government's military were actually conducted in the 2000s, their efforts to compromise U.S. properties in this so-called Operation Aurora included Adobe, Rackspace, Juniper, Yahoo, Symantec, Northrop Grumman, Morgan Stanley, and Google. All these attacks came to light in early 2010. So it was called Operation Aurora. And it marked a turning point for Google.

After Google discovered and publicly disclosed the attacks against its infrastructure, they decided to stop working with the Chinese government in censoring the search results for Google.cn. And Google eventually, as we know, shut down their operations in China. In explaining their decision, Google specifically mentioned Operation Aurora and the attack against their infrastructure by China as one of the factors behind their decision. And then, Leo...

Leo: Yes?

Steve: Wow, it does bring you back; doesn't it? The Sony PlayStation hack.

Leo: Oh, yeah. That was a big deal, yeah.

Steve: Yep. The Sony PlayStation attack, the spring of 2011.

Leo: Not the only one of the decade, either.

Steve: Yeah, announced that a hacker had stolen details for 77 million PlayStation Network users, including personally identifiable information and financial details. It's interesting that by today's measure such a breach would be a bit of a yawn. Oh, another massive breach of personal information. But at the time, and for many years afterwards, this breach stood alone as a biggie and really did impact them. It was catastrophic for Sony. They were forced, as a consequence of this hack, to shut down their cash cow, the Sony PlayStation Network, for 23 days. Remember it was off for, like, nearly a month.

Leo: Yeah.

Steve: While their IT people addressed the security breach. And it remains today the longest outage in the PlayStation Network's history. They not only lost profit directly due to the outage, but then more so over class-action lawsuits filed by users after some started noticing credit card fraud against the information that had been leaked during this breach. And then they still lost more money when they were forced to give users a bunch of free PlayStation 3 games as an incentive to bring them back. What the industry learned in a big way was the degree of damage that a successful network attack could cause when a company fails to invest in proper security. And of course we'll be talking about the Sony Pictures attack here in a little bit. There was another problem with Sony.

So this event also stands out because corporate lawyers woke up and started a trend of companies adding CYA clauses to their Terms of Service requiring that users relinquish any rights to lawsuits following security breaches. That wasn't in typical Terms of Services before the Sony breach. But now that is a standard of rights that are waived when users use websites and network services and so forth. It's like, well, breaches may happen. They're acts of god. So oops, you know, you have to hold us harmless for any consequences of those.

And in the latter half of 2011 was Diginotar.

Leo: Oh, Diginotar.

Steve: Just fun to say "Diginotar." That was when we first learned of the Iranian government's successful hack of the then-popular in that neck of the woods Dutch Certificate Authority, Diginotar. In a very nice retrospective summary which Threatpost wrote a year later - so that was in, I think it was in late 2012. Looking back on the event, Threatpost wrote: "The attacker who penetrated the Dutch Certificate Authority Diginotar last year had complete control over all eight of the company's certificate-issuing servers during the operation, and he may have also issued some rogue certificates that have not yet been identified. The final report from a security company commissioned to investigate the Diginotar attack shows that the compromise of the now-bankrupt certificate authority was much deeper than previously thought.

"In August 2011 indications began to emerge of a major compromise of a Certificate Authority in the Netherlands, and the details quickly revealed that the attack would have serious ramifications. The first public acknowledgement of the attack was the discovery of a large-scale man-in-the-middle attack waged against Gmail users in Iran. Researchers investigating that attack discovered that the operation was using a valid wildcard certificate issued by Diginotar for *.google.com," giving the attacker the ability to impersonate Google to any browser that trusted Diginotar's certificate, as they all did at the time.

Leo: What fun.

Steve: What?

Leo: Oh, just laughing. What fun. You can all be Google. You be Google, and you be. By the way, I did recommend NextDNS back in May. Somebody found a clip from This Week in Google. So I don't know how I found them, and I don't remember the conversation at all. But it was a Pick of the Week. It was one of those things where that must have just come out, and I thought, oh, here's something really cool.

Steve: It would have because they launched in May.

Leo: Yeah.

Steve: So you probably found a new DNS provider. And again, the fact that Mozilla has added them to their list of two, Cloudflare being the other one, it does both, yeah.

Leo: It's a good recommendation, yeah. So I take it back. I take it back, NextDNS. I did recommend you way back then.

Steve: So it quickly emerged as we looked at, I mean, the whole industry went, "Holy sh*t." Remember that, like *.google.com? So what we discovered was that attackers had also obtained valid certificates for a number of other high-value domains, including Yahoo, Mozilla, and others. Browser manufacturers immediately revoked the trust in the compromised certificates and reassured their users that the Internet was not broken.

Now, the final report from Fox-IT, which is the Dutch company that was brought in at the time of the attack in 2011 to find the root cause and determine the extent of the damage, says in its final report that the attack was a wide-ranging one that likely started more than a month before the CA themselves discovered it. And of course it was discovered when, as we recall, somebody used Google. Somebody used Chrome. Google has always been good about deliberately pinning its own certs. And so it was when someone used Gmail through Chrome that an alarm went off in Palo Alto headquarters saying, "A browser of ours just received a valid cert that we didn't produce from some outfit called Diginotar. What the heck is going on?" So that was the beginning of that drama.

And that ends 2012. How many times since 2013 have I referred to on this podcast Edward Snowden, and the fundamental way the Snowden revelations changed the world's entire security landscape? I think it would not be an overstatement to conclude that the Snowden leaks were probably the single most important cybersecurity event of this decade. They exposed, as we know now and didn't before, a global surveillance network that the U.S. and its Five Eyes partners had set up after the 9/11 attacks. And cybersecurity has been forever changed.

An unfortunate event, or another consequence of Edward's revelations, is that that drove the repressive countries like China, Russia, and Iran, to ramp up their own surveillance operations and to increase their foreign intelligence gathering efforts, which has led to an overall increase, an escalation in cyberespionage as a whole. And it was interesting. I did a little bit of browsing around. Wikipedia has really extensive coverage on the

downstream impact of Snowden's leaks. But, boy, you know, the Snowden revelations, well, I mean, as I said, and I have been saying, fundamentally changed.

Leo: If you had to pick one security event in the decade, that's the one.

Steve: Yes.

Leo: By far.

Steve: Yes. I mean, there were other big hacks. But this changed everything. I mean, just it changed the way we operate.

Leo: It was ironic because it wasn't anything we didn't already know. It just confirmed in the most dramatic way how much state surveillance was going on in the U.S.

Steve: Yeah. Yeah, exactly. And we had then visions of an - they were found, these network TAP rooms, where at the major exchange points on the Internet there was a dark closet, and all the fiber was passing through it. It was like, what?

Leo: And we knew that already from a whistleblower about AT&T. But again, somehow Snowden crystallized everything. And because the information was from the NSA, it had dramatic impact.

Steve: Yeah. And then at the end of 2013 Target admitted that malware planted on its stores' systems had enabled hackers to collect payment card details for 40 million of its previous shoppers.

Leo: That was that long ago.

Steve: Yeah.

Leo: That's still the prototypical breach.

Steve: It is. That's exactly right. And as a consequence, much of the world was introduced to the concept of point-of-sale malware, which wasn't something that we were really aware of at the time. And of course that meant that now consumers understood that purchasing something at an infected retailer could put their credit or their purchasing information at risk. There had been previous smaller incidents of point-of-sale malware. But this was the first time that a major retailer suffered a breach of such proportion. And we know that many retailers have fallen since. But exactly as you said, Leo, Target was the first biggie. And it still sort of remains like, ooh, you know, the Target breach is like, whoa, yeah.

Then in November of 2013 we had the big Adobe hack, where Adobe admitted that hackers had stolen the data of more than 153 million of their users. The data was dumped online, and their users' passwords were almost immediately reversed back to their plaintext versions because back then, 2013, heavy salting and much better PBKDFs were not in use at that point, or at least not at Adobe. And for many years after the Adobe breach, it continued to be used as a cautionary warning about the use of weak and easily guessed passwords. And it's worth noting that it was the Adobe breach that brings me to another major event, which actually wasn't until a little bit later. And that is, but it's significant, and that's Troy Hunt's creation of "Have I Been Pwned?" Troy we know is an Australian security researcher.

As a consequence of the Adobe breach, he created - and since unfortunately there was 153 million records of Adobe users - he created a simple way for users to go to Have I Been Pwned? website and learn, without disclosing any information because it was done client-side, whether their password matched the hash of a password that had been leaked. Anyway, Have I Been Pwned? marks another major milestone in this past decade. Today it's been such a success that Troy's site includes data breach databases - get this, Leo, this is an indication of the trouble - of over 410 hacked sites who have lost their data.

That's one of the reasons why the truth of SQRL being it gives no websites any secrets to keep is significant. The SQRL identity at a website doesn't do any other website or anybody else any good. So SQRL gives sites no secrets to keep, unlike these 410-plus hacked sites whose information totals more than nine billion hacked accounts is now available through hashing access at Troy's Have I Been Pwned? site.

Also, 2013 was the year that Silk Road, which was that Tor-based dark web marketplace for selling illegal content, basically drugs, people, everything, it got taken down. Silk Road's discovery and takedown for the first time showed the world, and certainly the security world, since we did a podcast before that about The Onion Router, TOR, the TOR network, and what a strong technology it had for protecting people's identity and privacy. But we now know, we learned with the Silk Road takedown, it isn't absolute; that it is not possible to provide perfect anonymity and security.

And 2014, the hack of Sony Pictures. It was in 2014 that we learned that North Korea had some competent hackers of their own.

Leo: Oh, yeah.

Steve: They initially called themselves the Guardians of Peace, and then later the Lazarus Squad. And they were eventually linked to North Korea's intelligence apparatus. As we'll recall, the motivation behind the hack was an attempt to force Sony to abandon its planned release of "The Interview," which was a comedy about an assassination plot against North Korea's then-leader Kim Jong Un. When Sony refused to be intimidated, the hackers damaged their network and leaked studio data and private emails online. And as our listeners will recall, this also was the first widespread use of the term APT, Advanced Persistent Threat, since we learned that Sony's breachers had been lurking inside their network for quite some time before. And a couple other events that we are about to get to will be reminding us of the notion of APT.

Also 2014 was the hack, the big hack of Mt. Gox. It was not the first cryptocurrency exchange to get hacked, but it remains to this day the biggest cyber heist of the cryptocurrency ecosystem. The hack, which is still shrouded in some mystery, occurred in early 2014. And Leo, are you sitting down? Are you centered over your ball?

Leo: I'm on my ball, yeah, centered, yeah, balancing, yeah.

Steve: The hackers made off with 850,000 bitcoins currently worth more than $6.3 billion.

Leo: Holy moly.

Steve: And we know...

Leo: Then they forgot their password. It's so sad.

Steve: Isn't that a tragedy. They're busy trying them all.

Leo: Yeah. We don't know who it was, though, do we.

Steve: No, we still don't know what happened.

Leo: Have those coins ever been cashed in? Or are they just sitting somewhere?

Steve: That's a good question. I have not looked.

Leo: Because you could look on the, you know, the thing about bitcoin is there's a registry.

Steve: Yes, there is an audit trail, yeah.

Leo: Yeah, and every single transaction's in there, and everybody, by the way, who has a bitcoin wallet has all of those transactions. So you could search for that number.

Steve: Incredible.

Leo: Wow.

Steve: Incredible. And at the time bitcoin wasn't as liquid as it is now. Now it's a freely exchanged commodity.

Leo: People would notice, though, if you sold $6 billion worth of bitcoin all of a sudden.

Steve: Yeah, actually the market would notice, yeah.

Leo: Yeah.

Steve: That would be an event. So unfortunately, since then, just as the infamous bank robber Willie Sutton explained that he robs banks because that's where the money is, hackers have since realized that stealing virtual currency is a lot easier than making it the old-fashioned way. So now cryptocurrency exchanges have become a frequent target of attack because that's where the virtual money is.

Leo: And Mt. Gox is long gone after that.

Steve: Yup. That's over. And no review of 2014 would be complete without reminding everyone of Heartbleed - that probably established the need to have a website and a kickass logo to go with your security vulnerability announcement, since that's now become de rigueur - a not-just-theoretical remote data extraction vulnerability that rocked the Internet at the time. Oh, baby, did we get some mileage out of that on this podcast, Leo. Heartbleed. It really did enable the discovery of a server's private keys with a low, but non-zero, probability.

There were a number of people who famously thought, well, okay, yeah, maybe. It was a buffer overrun, remember. We got a chunk of RAM out of the server. And the question was, okay, is there going to be anything useful in that RAM? And it turned out, yeah. Most of the time no, but maybe. And you could definitely get a buffer and start scrutinizing it. The promise of compromise was so juicy that it also began what is now the common practice of jumping immediately on top of newly announced vulnerabilities before they can be patched.

Heartbleed was exploited almost immediately after being publicly disclosed and led to a long string of attacks during 2014 and after. And even today some server operators fail to patch their OpenSSL instances, despite repeated warnings. At the time when it was publicly disclosed, it was believed that about half a million Internet servers were vulnerable. And that's a number which took many years to decline. And, boy, 2014 was a busy year for us.

Leo: By the way, you're having the same problem we had on TWiT on Sunday, where the decade seems to be weighted heavily towards the first half.

Steve: Yeah, it does, yeah.

Leo: You know? I don't know why that is. But yeah, we were stuck. It took us two hours to get to 2014.

Steve: Yeah.

Leo: I'm giving you a little pause so you can caffeinate. You can take a breath. We're not even halfway. Well, I guess we're sort of halfway there. What a decade this has been.

Steve: It has been, indeed. June 24th, 2014. The paper carried the innocent academic title, "Flipping Bits in Memory Without Accessing Them."

Leo: Oh, boy. I now know where we're going.

Steve: "An Experimental Study of DRAM Disturbance Errors." And the result was Rowhammer, another authentic, not just theoretical, attack. And of course we had Double Rowhammer, and we had Drammer and Rowdrammer and, I mean, it went on and on and on. As we know today, it would prove to be just the first of the many that we would be seeing in subsequent years, the many attacks against the computing hardware that we had naively believed up till then to be bulletproof. Eh, not so much.

And then, on the social engineering juicy side, 2015 was the year of the Ashley Madison dating website data breach.

Leo: Oh, yeah.

Steve: Uh-huh.

Leo: That one struck terror into the hearts of quite a few people.

Steve: It did. It was in July of 2015 that a hacking group calling themselves the Impact Team, and that was actually well named, released the internal database of Ashley Madison, which was a website aimed at those wishing to have a relationship outside of their marriage, an "affair," as they're called. Whereas most breaches today may expose our username and a hashed password for websites we may not even recall visiting, the Ashley Madison breach exposed people's real-world lives; and, sadly, a few committed suicide after being publicly outed as having an account on the site. So there was a breach that had some very real-world consequences.

Leo: Yeah. We've since learned that most of the women on Ashley Madison were just bots. It was almost all guys.

Steve: Yeah. Surprise.

Leo: Yeah, what a shock, huh?

Steve: So the practice known as SIM swapping, where hackers contact a mobile provider and trick the providers' personnel into transferring a victim phone number to a SIM card controlled by the attacker, first surfaced also in 2015. The initial SIM swapping attacks were linked to incidents where hackers reset passwords on social media accounts or grabbed a highly sought-after username. But as with Willie Sutton, once hackers realized that they could also use the technique to gain access to cryptocurrency or people's bank accounts from where they could steal large sums of money, the practice became much more of a nuisance, and there was a lot more pressure to authenticate that kind of attack.

It turns out that one of the problems is that our U.S. mobile carriers would prefer to do this online or over the phone, rather than requiring an in-person visit, as is typically required outside the U.S.

Leo: Really.

Steve: That's one of the - yes.

Leo: Oh, that's interesting.

Steve: That's one of the reasons that it was much more prevalent here is that you do some weak proof of identity using information which can also be hacked, and then you can escalate your attack.

Leo: They come to your house to deliver the SIM? Or you probably have to go to the store.

Steve: Yeah, you have to go in, in person, and show identity and so forth.

Leo: Sure. That's so much - why don't...

Steve: I know. I know.

Leo: We should really be doing that.

Steve: It's like banks these days. No one wants to see you.

Leo: Yeah.

Steve: It requires money to have a teller standing behind a counter somewhere. Also, in December of 2015, that's when the cyberattack on Ukraine's power grid occurred, causing power outages across Western Ukraine, making it the first successful attack on a power grid's control network that had been recorded. Although Stuxnet and Shamoon - that was a related attack - were the first cyberattacks against industrial targets, the Ukraine incident was the first one impacting the general public. It opened everyone's eyes to the dangers that cyberattacks could pose to a country's critical infrastructure. And as we know, the threat continues to loom today. We often hear of the idea of our own electrical grid being taken offline, and the presumption is, yeah, if bad guys wanted to do it, it's just not that well protected, unfortunately.

2016, it was the spring of 2016 that the Democratic National Committee, the DNC, admitted that it had suffered a security breach after a hacker going by the name of Guccifer 2.0 started publishing emails and documents from the organization's servers. It was later determined that the DNC had been hacked, not by one, but by two Russian bears, cyberespionage groups Fancy Bear (APT28) and Cozy Bear (APT29). The data that

was stolen during the hack was used, as we know, in a carefully staged intelligence operation with the intent of influencing the upcoming U.S. presidential election. So that DNC hack certainly ranks as one of the bigger events, a cybersecurity event for the U.S., at least, that occurred during the decade because it's continued to have repercussions in U.S. politics.

It was also in 2016 that Yahoo admitted that it had suffered two data breaches in the span of four months, including one that would turn out to be the largest breach in the history of the Internet. Whoopsie.

The Shadow Brokers. Although to this day we still don't know who they are, boy, did they have an impact. It was between August 2016 and April 2017 that the group calling themselves the Shadow Brokers first teased, then auctioned, and then leaked the hacking tools developed by the Equation Group, which we know is the codename for the U.S. National Security Agency, our NSA. These tools were top-quality hacking tools which made an immediate impact. A month after the final Shadow Brokers leak, one of those tools, EternalBlue, gave the WannaCry worm the teeth it used to wreak havoc across the global Internet.

It was a blog post in early September of 2016 which introduced the world to Mirai, a strain of Linux malware designed to work on routers and smart Internet of Things devices. During the 90 days that followed, after being used to launch some of the largest DDoS attacks ever seen, Mirai would become one of the most well-known malware strains in the world. And after its source code was released online in an attempt by its author to disavow its own authorship, it became one of today's most widespread malware families, with its code being the foundation of most IoT/DDoS botnets in use today.

This podcast has used two slogans as the result: "The 'S' in IoT stands for security," and "IoT is short for 'Installation of Trojan.'" And unfortunately IoT is going to be a challenge moving forward. And Leo, we should pause just for a second. I didn't pick up on the news of Apple and Google teaming up to produce an IoT security standard?

Leo: Isn't that wild?

Steve: Did you talk about that on...

Leo: Oh, we talked about it on TWiG, and we talked about it on MacBreak Weekly today.

Steve: Somehow I missed that.

Leo: Yeah, it's, well, you know how standards bodies are. We'll see.

Steve: It's an initiative. It's a committee.

Leo: And security is going to be a big part of it, by the way. But I think really what it is, is an acknowledgment that the IoT space can't really take off if you have...

Steve: It's too fragmented?

Leo: Yeah, so many protocols.

Steve: Ah.

Leo: So even Z-Wave in response announced they're going to open source Z-Wave.

Steve: And so interoperability is the goal.

Leo: Yes, exactly.

Steve: Ah, okay. Well, that'll be really nice, too.

Leo: I hope, if that happens. But I have my doubts. I think you get these big companies sitting down at the table, they're going to say, "Well, I want this." "Well, no, you can't have that. I want this." We'll see if they actually ever agree on anything. We'll see. I hope so.

Steve: So it was May 2017 that WannaCry first swept across the Internet, which was as we know fueled by the Shadow Brokers leak of the NSA's EternalBlue exploit against Windows file sharing. We now know that WannaCry was also developed by North Korean hackers looking to infect companies and extort ransom payments as part of an operation to raise funds for the sanctioned Pyongyang regime. So okay, way to make money, guys. Yeah, wow.

Vault 7 was WikiLeaks' last good leak. It was a trove of documentation files describing the CIA's cyber weapons, sort of the equivalent on the CIA side of the Shadow Brokers leak over on the NSA side. No source code was ever included; however, the leak provided a look into the CIA's technical capabilities, some of which included tools to hack iPhones, all the major desktop operating systems, the major browsers, and even smart TVs. At the time, WikiLeaks said it received the Vault 7 data trove from a whistleblower. Later he was identified as Joshua Adam Schulte. So that was also in '17.

And then the MongoDB. Although incautious sysadmins had been leaving databases exposed online without any password for quite some while - after all, iCloud, the cloud in general had already happened - 2017 was the year when hackers finally turned their attention to this previously untapped Internet resource. It began in the tail end of 2016 and really picked up steam by January of 2017 with hackers accessing databases, deleting their content, and leaving ransom notes asking for cryptocurrency to return the nonexistent and previously deleted data.

Although the first wave of attacks targeted what was essentially low-hanging fruit of MongoDB servers, hackers later expanded their reach to encompass other database technologies including MySQL, Cassandra, Hadoop, Elasticsearch, and others. Although widespread attacks died out by the end of 2017, they served to highlight the significant dangers of publicly exposed misconfigured databases.

Leo: You aren't used to talking for so long without a break, are you.

Steve: No.

Leo: You want me to do a phony ad?

Steve: No, it's good. Got a piece of phlegm.

Leo: Oh, yeah. I know how that feels.

Steve: Yeah. Anyway, so I'm still talking about what a problem these databases are. By the end of the year, a new category of researchers known as "breach hunters" had been born. These were individuals who looked for open databases and then contacted companies to let them know they're exposing sensitive information online. Oh, by the way, I just thought you'd like to know. During 2018 and 2019 most security breaches and data exposures are now being discovered by breach hunters, rather than hackers dumping a company's data online after an intrusion.

Leo: That's good.

Steve: So it's nice that we have breach hunters. It's dumb that all these databases just keep being left exposed online. It's like, dummies. Yeah.

And 2017 was also the year of the Equifax hack, in which the personal details of more than 145.5 million American, British, and Canadian citizens were stolen from the company's systems. We know that the breach was the fault of Equifax failing to patch the Apache Struts vulnerability for about 90 days. I think the breach occurred about 90 days after Apache Struts' vulnerability was announced. And they just hadn't gotten around to it. We still don't know who was behind the intrusion, nor what their motives may have been. But it was a big lesson about the need to patch because this was of course extremely expensive in terms of reputation damage and real-world lawsuits against Equifax.

And then what would 2017 be, Leo, without Coinhive? It was the latter half of 2017 that Coinhive appeared, and the term "cryptojacking" was coined. The Coinhive service, as we know, enabled hackers to make money by mining the Monero cryptocurrency on other people's computers after somehow arranging to load a snippet of JavaScript into a victim's browser. If nothing else, just putting an ad up, just buying an ad that had this JavaScript in it. When the ad was on someone's browser, their machine would be busily part of a mining pool and generate a little incremental piece of a Monero coin.

And for a while during that crazy, like, bitcoin was at $20,000 per coin, during that crazy peak it made sense. Unfortunately, nobody liked having other people using their computers to mine coin, so the browsers actively pushed back on the practice. And then after the collapse of cryptocurrency valuations, it rendered the whole scheme much less profitable. And of course, as we know, it was later in 2018, I think it was 2018, that Coinhive shut their doors. And then, as we covered at the time, it wasn't long before somebody else jumped in to offer Monero mining services to the bad guys.

Coming up with the big security event of 2018 is a no-brainer when you have the likes of Meltdown, Spectre, and the myriad other attacks which were subsequently discovered to be theoretically effective against our modern processors' speculative executing microarchitectures. Even though not a single one of these was ever found to be exploited

in the wild, by the time we had learned that, given time, well, by the time we had learned - I got my phraseology here in what I wrote doesn't make sense to me.

Even though not a single one of these was ever found to be exploited in the wild - well, I still don't know what I'm trying to say here. By the time we had learned that - oh, I see. By the time we learned that, given time and sufficient motivation, bad guys will arrange to leverage any - oh, I see. By this time we had learned that, given time and sufficient motivation, bad guys will arrange to leverage any perceived weakness into a working exploit. So every single one of these fundamental microarchitectural mistakes needed to be, and have been so far, cleaned up. Future CPU design has been hugely impacted as a result, at the cost of some hopefully short-term performance.

And Leo, you've often said this, and I concur completely. It is a little bemusing that, yeah, all of this hay has been made of Spectre and Meltdown and all their ilk during the last really full two years, 2018 and 2019. Yet as far as we know, not a single instance of these vulnerabilities has ever actually been used against anyone. Maybe it would have been an attack that allowed VM boundaries to be breached, though even then it was only theoretical. I mean, you know, it was demonstrated. Proof of concepts were demonstrated. So, yeah, you don't want that to be in a cloud system where people who you don't trust are able to run code on your shared hardware. But it certainly never had any impact on end users, despite the fact that end users have suffered a performance hit. So anyway, clearly the event, the security event that stands out for 2018 was Meltdown, Spectre, and all of the other speculative execution attacks.

And this was also when we learned of the Starwood/Marriott data breach. Though not as big as Yahoo's three billion figure, the Starwood/Marriott data breaches deserve a mention, if for no other reason due to its sheer size and the fact that a lot of personal information, like passport information, was lost. The breach, which was disclosed in November of 2018, impacted around 383 million of the hotel chain's guests. At the time, the first estimate you'll remember was 500 million. But after further forensic investigation they said more like 383 million. Forensic investigators dug into the Starwood reservation system, and they discovered a RAT, a Remote - they found a rat in the system - a Remote Access Trojan and the Mimikatz tool, which is used to sniff usernames and passwords.

There's never been any public disclosure of the route into the Starwood systems. So it's entirely possible that only the attackers know how they originally got in. But this was, again, another instance of an APT, an Advanced Persistent Threat, because those guys were there for quite a while, exfiltrating all of this data. And in fact remember it was only when an auditing tool happened to be on the network that saw an odd query of the database that tripped a filter that said "here's an odd query." And then when it was looked into more deeply, the username of the query was not one that was online or present or something at the time. So then they dug deep, and they ended up saying, oh, crap. Not only do we have a big problem, we've had a big problem for quite some time.

And I will wrap this up with 2019, just declaring it the year of the ransomware. I think that has been really the overarching story of the year because bad guys have figured out that they can encrypt data on exposed systems. There's now huge pressure to get into systems, to encrypt data, and make that happen. We also have this new sort of multilevel marketing model that GandCrab pioneered where the bad guys syndicate the attack, essentially, by giving the syndicatees, or syndicators, I guess, yeah, the syndicators - would it be "ees" or "ors"? The person who syndicates is a syndicator.

Leo: The content provider is the syndicator. The content receiver is the syndicatee? I don't know.

Steve: Yes. Yeah, I think so. I think that's right.

Leo: I think neither word makes any sense. It sounds like gobbledygook to me.

Steve: Anyway, the point is that, as we know because we've talked about it a lot, the idea is the bad guys behind GandCrab who initiated this get a whole bunch of minions to go install the GandCrab malware; get it into other people's systems; and, very smartly, give them the lion's share of the ransom, which keeps the minions incented to infect other people's systems with this malware. Meanwhile, the GandCrab guys deal with all of the backend. They negotiate...

Leo: It's more like a franchisor and a franchisee.

Steve: Yeah, I guess you're right. It's a franchise, yes.

Leo: Yes, a malware franchise. Be the first on your block.

Steve: And that's 2019 in a nutshell. As a consequence of that, we saw a huge number of attacks throughout the year, ransomware attacks. And the problem is, since much of the way in is social engineering, and that's unfortunately not something that technology can completely solve, I have a feeling we're going to be seeing more of that in 2020.

Leo: Yeah.

Steve: And there, Leo, is our decade in review.

Leo: That's the decade. You heard it here. All 10 years in review. What do you think 2020's going to look like? It's definitely ransomware.

Steve: I think it's ransomware, and I think it's IoT. I think that after Christmas - this is going to be the IoT Christmas. And we're going to...

Leo: Oh, you're right.

Steve: And it's going to be the IoT death in January.

Leo: Oh, you're right. Very interesting. One thing that, you know, if I look at the decade from the overview, that most impresses me is the ingenuity of hackers and their willingness, like water, to find whatever cracks there are. They just flow; you know?

Steve: And Leo, some of these attacks are phenomenally impressive.

Leo: Pretty sophisticated. Yeah, yeah.

Steve: Oh, my god. I mean, if I, like, ever read every science fiction book so that there were no more, and the Internet - well, okay, wait, it can't go away because then I wouldn't have it to - the point is, way down on my list is having the time to invest in reverse engineering other people's work in order to find flaws. Because, I mean, I like to read. I like science fiction. And there's TV, and there's movies, and there's podcasts. It's like, you know, all this other stuff. But, boy, I am so impressed with the work that these hackers and attackers do. It is, I mean, if they pointed their skills to being productive, rather than being in the dark underbelly, we'd have, like, IoT security and things that would be really useful. But no.

Leo: But no. Steve, what a year it has been. What a decade it has been. It is always a pleasure as we head into our third decade of the shows. I always look forward to our Tuesday afternoons together, 1:30 Pacific, 4:30 Eastern, 21:30 UTC. That's when we do Security Now!. You can watch us do it live at TWiT.tv/live. Today's show we did a day early because we didn't want to work on Christmas Eve.

Steve: Better than a day late.

Leo: Yeah. And next week's show is New Year's Eve, and we won't be doing a show then. We'll be doing a "best of." And we've got the highlights of 2019 all put together by our producers and our editors. But you and I will reconvene in the new year. Actually, I won't be here in the first Tuesday of the new year, which is...

Steve: Really.

Leo: Yeah, I'm going to be in Vegas for the Consumer Electronics Show.

Steve: Oh, CES, baby, CES.

Leo: So we'll be back January 7th, but it'll be probably Jason Howell. And then I'll be back with you on the 14th.

Steve: Well, you know what I'm watching, Leo.

Leo: What are you watching?

Steve: I'm watching the count of Security Now! episodes because we're now at 746. In four more we'll be at 750.

Leo: Yeah?

Steve: And then we start the final quarter.

Leo: It's the final countdown.

Steve: The wind down, the final - although it's not like it's going to be happening fast, the final 250 episodes.

Leo: Take a few years.

Steve: Yeah, that's right, another five years, that ought to do it. And by that time IoT will be secure. Ransom will have been cured. Really, we'll be just like, you know, we'll have - well, you'll still have hair.

Leo: By that time it'll take two people, three people, maybe four people to do the show just to keep up.

Steve: Perfect. Then I will happily hand it off to them. But not to worry. Not for another five years, everybody.

Leo: Whew. Steve, always a pleasure and an honor to work with you. It's been a great year. Thank you for the - I know you put so much work into this, and I really - we're all very grateful that you do. It's one of our most popular shows, and people just love hearing what's going on from you. I have a fellow in the studio, he came here for the show today. He works in security at a major social network. Actually, not exactly security. He's a developer. But security is one of your issues, absolutely.

Steve: Has to be these days, yeah.

Leo: He listens to the show to make sure that he's up on everything going on.

Steve: Very cool.

Leo: Well, have a wonderful holiday.

Steve: Okay, my friend. Have a great holiday and a New Year. And try to not walk too much during CES. I know those are exhausting shows.

Leo: I'll try to come back healthy. That's going to be the biggest trick.

Steve: And I love 2020. I just like the shape of that.

Leo: Isn't that a great year? Yeah.

Steve: 2020 looks like a great year.

Leo: Feels like the future.

Steve: With any luck that'll be the kind of vision we have: 2020.

Leo: Nice. Thank you, Steve. GRC.com. That's his website. Get the show there or at TWiT.tv/sn. Subscribe. And don't forget SpinRite, the world's best hard drive recovery and maintenance utility.

Steve: And just for our listeners, those who've hung on here to the very end, I am deep into bringing up the new Unix server and moving - I'm now in the challenge of moving the newsgroups over to the new hardware.

Leo: XenForo runs on Windows only; right?

Steve: No, it's a PHP.

Leo: Okay, so anywhere.

Steve: But these are not the SQRL forums.

Leo: Oh, okay.

Steve: This is the old-school NNTP. You've got to have a news reader-style newsgroup. That's where all the real work happens at GRC is really deep under.

Leo: That's good. It keeps the plebeians out.

Steve: It does. It was funny, we used to have - I had a browser, I do have a read-only browser interface. And it used to be read-write. And we had these problems with just all this crap being posted in the newsgroups. And I don't think I can take credit for it, but somebody noted that all the crap was coming in through the browser interface. And I said, oh, problem solved. I just set it to read-only, and it was much better.

Leo: Much better.

Steve: Much better.

Leo: Much mo' betta.

Steve: Okay, my friend. Next year.

Leo: Thank you, Steve. See you next year.

Steve: We'll do it again. Bye.

Copyright (c) 2014 by Steve Gibson and Leo Laporte. SOME RIGHTS RESERVED

This work is licensed for the good of the Internet Community under the Creative Commons License v2.5. See the following Web page for details: http://creativecommons.org/licenses/by-nc-sa/2.5/