SonarSnoop

Description: This week we cover the expected exploitation of the most recent Apache Struts vulnerability, a temporary interim patch for the Windows zero-day privilege elevation, an information disclosure vulnerability in all Android devices, Instagram's moves to tighten things up, another OpenSSH information disclosure problem, an unexpected outcome of the GDPR legislation and sky-high fines, the return of the Misfortune Cookie, many thousands of Magneto commerce sites being exploited, a fundamental design flaw in the TPM v2.0 spec, trouble with MITRE's CVE service, Mozilla's welcome plans to further control tracking, a gratuitous round of Win10 patches from Microsoft - and a working sonar system which tracks smartphone finger movements!

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-679.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-679-lq.mp3

SHOW TEASE: It's time for Security Now! with Steve Gibson. I'm here in place of Leo Laporte, so I'll do my best to keep up. We're going to be talking all about some updates on the Apache Struts vulnerability from last week. There's a new vulnerability in all Android devices that aren't running Pie which is basically all Android devices. A very unexpected outcome of the GDPR legislation in the EU. And we're going to talk about something called SonarSnoop, and it's really kind of strange and interesting. You won't want to miss it. Security Now! is next.

JASON HOWELL: This is Security Now! with Steve Gibson, Episode 679, recorded on Tuesday, September 4th, 2018: SonarSnoop.

It's time for Security Now!, the show where we talk about all the latest security news, everything that's happening on the security front. Leo is out for the next couple of weeks, actually.

Steve Gibson: Three.

JASON: This week and the next two weeks; right?

Steve: Yup.

JASON: So three solid weeks you're going to have to listen to me going, "What did Steve just say?" That's basically what my role is on Security Now!. Steve Gibson, you are the man with all the knowledge. How are you doing, Steve?

Steve: Hey, Jason. Great to be with you for the first of three. And you did this, what, about a month and a half ago, I guess.

JASON: Yeah.

Steve: And it went well.

JASON: Yeah.

Steve: And so we sort of have our routine down. I think it's going to be great.

JASON: I think it's going to be good. I will almost undoubtedly play the role of person asking you to - maybe not asking you to repeat yourself, but having a very large question mark above my head at the beginning of the story, and maybe a checkmark at the end of the story. It'll be like, ah.

Steve: Well, I guess since yesterday was Labor Day, this marks the official beginning of, well, is it winter? At least it's end of summer.

JASON: It feels, yeah, it feels like fall-ish.

Steve: As we plow into September; right? So off we go. So the most interesting piece of, well, the most interesting thing that happened in the last week, I think. Sometimes, as the last time you and I were together, it was just - there wasn't anything that really stood out. It was just like lots of news. This time there was, I mean, it was easy to pick one because some researchers also showed a paper, now it's been about three weeks ago, the USENIX security symposium in Baltimore, where they surprised everybody by figuring out how to use the colocated microphones and speakers on, in this case it was a Samsung S4 smartphone, to essentially disambiguate the unlock strokes that the phone user was using in what they called "SonarSnoop."

So it actually turns the microphone and speakers, or microphones and speakers because there's two of each, into working sonar in order to allow them to track the person's finger movements on the screen. So a really interesting piece of research there. And to me it feels like, because I read the PDF, and I think it was 18 pages of rather detailed research, I think this is something that, if anyone is interested, can be much further refined. But anyway, we'll get to that.

We did have a bunch of stuff. As expected, and as happened previously with an Apache Struts vulnerability, we have now seen lots of action on it in the past week, which we'll talk about. We've got an interim patch for last week's announced zero-day privilege elevation surprise. An information disclosure vulnerability present in all - I wrote here in my notes "all Android," but actually it's all before Android 9, a.k.a. Android P. Did that ever get an official name, by the way?

JASON: Yeah, it's Pie. It's Android Pie.

Steve: Okay, Pie. Okay, good.

JASON: And, I mean, really, those two are distinctions without much of a difference. How many people really have Pie right now? It's a pretty small percentage.

Steve: Yeah, because I had in my notes "pre-P." And I thought, what? No. So now it's pre-Pie.

JASON: Right.

Steve: We also have Instagram, after a bunch of high-profile hacks, has finally, like, okay, reluctantly it seems, moved to tighten things up. We'll talk about that. Another

OpenSSH information disclosure problem which is being handled differently than the last one in an interesting fashion. We've got an unexpected outcome of the GDPR legislation that took effect three months ago in the EU, not anything I would have predicted, but we'll talk about that. Also the return of something we discussed in 2014 called the Misfortune Cookie that affects medical devices. Also many thousands, I mean many thousands of Magento commerce sites are being exploited with a skimming script. We also have, believe it or not, a fundamental design flaw in the Trusted Platform Module v2 spec, which is what all of us now have, certainly in corporate servers and desktops. And many standard desktops have TPM protection in them in order to secure the boot process. Turns out that can be circumvented, so we'll talk about that.

Also we often, well, we're always referring to CVE numbers. It turns out that MITRE is the contractor, kind of of the U.S. government, that is running the system. And they've been having problems lately which maybe the government is going to come to their aid. Mozilla has announced plans to improve tracking control and some other aspects of Firefox. And we've got a gratuitous round of Win10 patches from Microsoft which actually my machine, my Win10 machine which I run Skype on, managed to get through this morning. I always deliberately start it hours in advance in order to let it get this out of its system beforehand. So that's done. And then we're going to wrap up with a discussion of a new side channel attack, bouncing sound waves off of the user's finger as they unlock their phone. So I think another great podcast for our listeners.

JASON: Absolutely. It's jam-packed and full. I've got to say real quick you're a brave man to do your Windows updates before the show. Even if it's hours in advance. Maybe that's different now. But I almost always feel like I have to push those off until after the show because you just never know.

Steve: Well, if you get behind it could be really bad.

JASON: Oh, for sure.

Steve: Like I went to a conference a year ago where someone had a laptop that I kind of got the sense they're normally deskbound, and so they're using their normal probably desk system. But they grabbed their laptop, and off they went to the conference. They plugged it in and turned it on. And it was like half a day. I mean, they were, like, sitting there not able to do anything while it caught up with everything that had gone on while it hadn't been alive. So if you bring Windows up every week or two, you're probably not going to fall too far behind.

JASON: Yeah, right. And don't get caught using conference WiFi to make all those updates. That's going to take you even longer.

Steve: Yeah, that's true. That's true.

JASON: That's like nightmare on top of nightmare.

Steve: That, too. That, too.

JASON: Well, we've got a lot of that stuff to talk about. Obviously a bunch of security news to kick off with. All right. So we've got a stack of security news here. And apparently you made a prediction. Did you make this prediction last week, as far as Apache Struts is concerned?

Steve: No. Well, we do need to first look at our Picture of the Week.

JASON: Okay. Let's do Picture of the Week.

Steve: It's one of those where you really can't make this up. I just got a kick out of this. Someone sent this to me, and I've had it in my list of pictures that we would get to when we had an opening, when there wasn't something else happening. So this is a screenshot that appears authentic, I have no reason to doubt it, which is the Windows error reporting dialogue, which has been presented to someone announcing that Windows Error Reporting has stopped working.

JASON: Oh, boy. How is it able to do that, if it's stopped working?

Steve: I guess it maybe squeezed this in just before it died.

JASON: It was its last breath.

Steve: No, that wouldn't work because it would say that it was stopped working, and then it was going to try to show you the dialogue. So I don't know. This is very fancy coding on Microsoft's part. Apparently it's able to anticipate the fact that it is about to die, and so let's show them a dialogue saying goodbye and goodnight as our last act. I don't know.

JASON: Oh, boy. Yeah. Seems a little fishy to me.

Steve: This is just so good. It's - yeah.

JASON: Or maybe once it crashed - because you can't anticipate a crash necessarily. But you can look backwards once it relaunches and say a crash happened. So maybe it pops it up when it relaunches somehow. Or I don't know how it's able to do that. That's very strange.

Steve: Yeah, well, and it says in the fine print a problem caused the program, in this case Windows Error Reporting, to stop working correctly. So I guess that means to start working incorrectly. Windows will close the program and notify you if a solution is available.

JASON: Oh, so in that case it hasn't actually closed it yet. So that's how it was able to put it up. Because if that was the program, then it's saying here's the error message, we're going to close it now. And so when you...

Steve: Okay. Or I guess maybe technically it could be reworded. Windows Error Reporting is about to stop working.

JASON: Right, exactly.

Steve: And so it's like, oh, here we go, this last chance.

JASON: And there's no button for "No, don't stop working."

Steve: There is always the power button.

JASON: That's true.

Steve: So, okay. Prediction. Yes. I did predict it, only because of what we recently saw. So Apache Struts is a very popular Java server-side application framework which is notably used by, I think it's 70-some, I don't remember the number now, but like a lot of the Fortune 100 major corporations in the U.S. It's just very popular. And it's had problems. It is the reason that Equifax had that breach that ended up costing them $600 million last year. What happened was there had been like six months before a notice of a

serious problem with the Apache Struts framework which allowed unauthenticated remote access. Equifax, one of those big corporations that uses Apache Struts, they just never got around to updating their Apache Struts, which allowed bad guys in, and their entire database got exfiltrated. So the industry hopefully took this as a wakeup call.

We did have, about three months ago I'm guessing, a different problem found. And, oh, I remember the way it was. It was on a Wednesday of whatever week it was. The announcement was, from the Apache Struts guys, a supercritical vulnerability is going to be fixed, patched a week from now, at some exactly certain time of the morning. So everybody, I mean, this is bad. So if you are affected by this, get ready. You're going to want to update your system immediately because, the developers wrote, we don't think it's going to take anybody long to reverse-engineer what we've fixed and then start scanning the Internet for vulnerable systems. It took hours before the vulnerability was understood, scanning began, and I think it was like three days before we began to see malicious attacks based on it.

So based on that, when we last week covered the fact that there was, whoops, another problem, not quite so bad, because the default, the vanilla installation of Apache Struts was not vulnerable. Now we know more about it. And there's a package known as Struts Convention which is a plugin that by default flips some switches to true, which enables this vulnerability. So the guy who found it last week explained that all Struts were vulnerable unless patched. And that's from versions 2.3 to 2.3.34, and on the 2.5 channel from 2.5 to 2.5.16, which are the ones that are actively in use and supported now. They're all vulnerable. So the discoverer of this explained that, even if you weren't now vulnerable, a configuration change downstream could make you vulnerable.

Now we know, we have all the details about that. The guys from Palo Alto Networks - I think that was Palo Alto Networks. I'm looking for it. I had it in my notes. Anyway, yeah, Unit 42 at Palo Alto Networks. They explained in detail what was necessary and also connected this to this Struts Convention plugin which would by default make any site using it vulnerable. So the guy who found it last week said, even if you're not vulnerable today, really update. And as we said last week, boy, given what happened to Equifax, unless you've got an extra 600 billion, I mean million, lying around, you probably wouldn't have that much exposure, but still worth updating yourself.

And so the prediction part is that, sure enough, within, well, actually what Palo Alto Networks said in their note was they said: "Some have noted that a previous critical Struts vulnerability was actively attacked last year only three days after the release of the security update and vulnerability information." And they continued, saying: "There are no known active attacks at this time." And, by the way, this was being written two days after the vulnerability disclosure. "And the current requirement that two non-default conditions need to be met for the vulnerability to be exploitable makes for a different threat environment." In other words, less concern, to some degree.

However, the vulnerability was disclosed on August 22nd. They wrote that on the 24th. Active scanning for vulnerable systems was observed on the 25th. Two cybersecurity firms, GreyNoise Intelligence and Volexity, have detected threat actors scanning for Struts servers last week, but they did not identify any attempts of exploitation. And as we know - and again, this pattern is repeating. So scanning precedes attacks. Attacks are expected. And there has been now, since then, some observation of coin mining being installed on vulnerable servers.

However, the most fun about this whole thing, and I'm hoping you're going to be able to put this on the screen, is that, for all time, the bar has been raised on vulnerability disclosure and demo sites. Our listeners will certainly remember the Heartbleed problem from last year. It had a great logo, this notion of, oh, vulnerabilities need good names, and they need good logos. And of course we have Spectre, the little ghost holding the

stick and so forth. Anyway, I have a link in the show notes to the Apache Struts CVE, and this is CVE-2018-11776.

JASON: Oh, yeah.

Steve: You're now showing it in the video stream. And it's like, okay. This has, for all time, upped the ante on a page which discloses the vulnerability and details it. This actually has a complete tutorial on the exploitation of the vulnerability. So anyway, it's becoming a little bit of a PR competition now when people come up with new vulnerabilities.

JASON: I think the next step is that they're going to embed a full 22-minute cartoon that explains. That's the obvious next step.

Steve: Wow. So also last week we had a person who - I just used the pronoun "he," not knowing any better. SandboxEscaper was the person's Twitter handle. And I just said "he" without knowing. I later picked up some feedback saying that this was actually a woman. I don't know - a female. I don't know how I would be expected to know that, but somebody did.

So for what it's worth, she tweeted, it was sad, to the effect that she really didn't care what happened to her in life any longer. She had found a vulnerability in Windows, Windows 10 64-bit, which allowed an attacker with no privilege to escalate or elevate to full system privilege. And she didn't feel like reporting it for whatever reason to Microsoft, so she just told the whole world about it, posted the proof of concept on GitHub and said here you go.

So, now, this is not super critical. It's not a remote code execution zero-day, where suddenly every Windows 10 system on the Internet would be vulnerable. But still, a privilege elevation, I mean, the reason we have privilege controls on all modern operating systems is we create this notion of a root user who can do anything and a typical user who administratively and deliberately limits what they can do so that, if a program there running on their account happens to try to get up to some mischief, it won't be able to go rewrite the OS files and so forth. So the point is that, while by itself it isn't super crucial, it is absolutely something that any malware would want in its toolkit because, until fixed, it allows such a program to do whatever it wants such that even the guest account, even a program running as guest is able to do anything it wants to on the system.

So now we know much more detail about it. It's been vetted by a number of other researchers. We discussed one last week that had already confirmed that this thing was legitimate. Since then a small tweak has been found that widens its scope to 32-bit systems, as well. And due to a hardcoded filename, it was Win10 only. Changing a digit 3 to a digit 1 allows it to also attack Windows 7. So its scope has broadened.

The good news is it's such a simple oversight in one API. This actually is in the Task Scheduler. The Windows Task Scheduler API has a function, Schedule RPC, which is Remote Procedure Call Set Security [SchRpcSetSecurity]. That function fails to check permissions. So anybody, even a guest who has deliberately restricted access rights, can call it and use it to set file permissions on any local file. The way the exploit works is it allows a hard link to be created and then calls a print job using the XPS printer, which ever since Windows XP Service Pack 2 has been present by default in Windows systems. That allows the spooler process to invoke a hijack DLL which has been given full system privileges. And then you're off to the races, and this thing can do anything it wants to on your system.

So it's expected that Microsoft will fix this easily. There is a company, Acros Security, which has been previously publishing a series of what they call "micropatches," which are basically little code tweaks to existing programs which may not be updated by their publishers because the publishers don't care. The software has gone out of support or whatever. They have produced one of these micropatches for this problem. On the other hand we're now - it's September 4th, the first Tuesday of the month, Patch Tuesday being next week. So within a week this is almost certainly going to get fixed by Microsoft.

And again, it doesn't expose you to badness from the outside. It's not network exploitable. It's purely a local API call requiring code running on your system. Again, it's not nothing. I mean, code running on your system which is malicious would love to have this. So it's not good that this SandboxEscaper person chose not to responsibly disclose this to Microsoft. But that's the way it is.

So there is a micropatch. It's just 0patch.com, if anyone is concerned and interested. I don't like the idea of applying third-party things to Windows. It turns out it was a very simple thing to fix. They moved a couple function calls around. Three instructions needed to be changed in order to fix this. So as patches go, it's probably about as benign as possible. But you're better off keeping bad stuff out of your system to start with. So anyway, that sort of wraps up - we'll wrap it up for sure next week when we verify that this has been fixed in next Tuesday's Patch Tuesday. But again, that sort of provides some more details to what we were talking about last week.

JASON: And how often does that happen where someone finds a vulnerability like this and doesn't go through the responsible disclosure of it and just drops it like a bomb onto the Internet in this way?

Steve: It's true. We're seeing this sort of responsible disclosure now has become the regular routine. Microsoft acknowledges the people who found it. All of the security papers that we talk about are being released only after the problems they describe have been fixed. So it's certainly more dramatic if you disclose some horror that still exists. But, I mean, you really then become an outcast within the security community. No responsible researcher will do that because you're just opening the floodgates for instantaneous exploitation.

I mean, even a perfect example is this Apache Struts problem. Here, it was responsibly disclosed. No one knew about it until the patch was made available. What we now know is that even things that are patched, often, I mean, even things for which patches are available still end up with systems for a long time remaining unpatched. So this is really the only way to go. It's only fringe researchers who just don't really care about their own reputation who do this. And that's sort of what she said in her tweet was "I don't care about life anymore." It's like, uh, okay. Sorry about that. But, you know. Anyway. And speaking of things that will never be patched...

JASON: You're talking about Android, aren't you.

Steve: I am. I'm sorry, Jason. I know it's your beloved platform.

JASON: You know what, for years I've been involved with Android. I've just gotten used to it at this point. It's just the way it is. Another vulnerability.

Steve: Just wear your armor. So all versions of Android running on all devices are believed to be affected because what we're going to describe here is a core Android function which Google has fixed in the most recent Android Pie release, but has said they have no plans to fix in older versions. And they're saying you're encouraged to upgrade to Android Pie or later. Well, I don't have to tell you, Jason, the likelihood of that

happening. I mean, like, in many places you can't. I mean, it's just not an option. You're stuck with the Android that you have. And the Android that you have, if it predates Pie, is subject to a CVE - this is the Common Vulnerabilities and Exploits database - 2018-9489, which has been assigned to this problem. And all Android devices prior to Pie, until they're updated, or maybe there'll be a patch, we'll see.

So what's going on? The OS broadcasts on purpose system messages which are global and which any application can indicate - they're called "intents." And any application can raise its hand and say, I'd like to receive those, please. And it then receives them without any user permission oversight. So these are not things like permission to use the camera or other major aspects of the system. These are sort of regarded as internal stuff that's going on.

Unfortunately, the messages include the WiFi network name to which the device is currently associated, the BSSID, the local IP address, the DNS server information, and the device's own static MAC address. Some of this information such as MAC addresses has long since been recognized as sensitive, and it has no longer been made available through standard Android APIs since Android 6. But now we're at 9, and even before 6, 7, and 8, that MAC address is available via the broadcast. So this argues that Google masking the MAC address wasn't fully done. The API to request it, it was removed from. But you can still raise your hand and say broadcast it to me, and then you receive it. So by listening to these broadcasts which are continuous, any application on the device can capture this information which in turn has the effect of bypassing all permission checks and existing mitigations.

So this leakage does undermine Android's system of permissions. Of course we know that MAC addresses do not change and are tied to the hardware. So this can be used to uniquely identify and track any Android device, even when MAC address randomization is used because, as we've discussed here, MAC address randomization is the pre-access point association MAC. It was realized some time ago, and Android and iOS both fixed this, and I guess Windows did, too, that wandering around just anywhere, WiFi is so present that it's just you're bathed in WiFi now. So it turns out that any access point could see the MAC addresses of all, does see the MAC addresses of every WiFi-enabled thing within its reach, even those that don't associate with it, that never have and don't. So what we realized was, okay, that's not so good.

So MAC address randomization causes the client to just generate an arbitrary random MAC address during the time that it is not associated. But when it associates, only then do you get the true MAC address. So that's sort of a nice privacy tradeoff that works, and which everybody is doing. The problem is that any app running in Android prior to 9 is able to determine the physical unrandomized MAC address, the network name, and the BSSID. And of course there are databases. There's WiGLE, W-I-G-L-E, and Skyhook, both which are comprehensive databases of exactly that information, which would allow software to geolocate the user using that information. Even when that software has not been granted any location permission, when location is completely shut off on the device, presumably because the user wants privacy, this is still disclosing it. So unless some sort of mitigation is made available, this is the way it's going to be on all Android prior to 9. There's a cool website, I thought I had it here, the website.

JASON: Are you talking about the app or...

Steve: Yeah, the Android broadcast monitor logger.

JASON: Yeah, broadcast logger or - let's see here. Internal Broadcasts Monitor?

Steve: Yes. It was, yes. Internal Broadcasts Monitor is the application, developed by a developer who also put the source on GitHub. Internal Broadcasts Monitor is available

through Google Play. And so you can install it from the Play Store. You tap Start, and you start observing all of the traffic which is available to it, not having obtained any permissions whatsoever. You look for android.net.wifi.STATE_CHANGE and android.net.wifi.p2p.THIS_DEVICE_CHANGED messages, and the data they carry, and you will see all this.

So not a huge problem. But if a user is interested in not being located, the only way to do that before 9 and until this is maybe fixed by a third-party patch, or maybe Google will reconsider their position - but on the other hand there's all these phones that are not Google and that are not being patched - would be to turn off WiFi because basically the software is still going to know your MAC address. Well, okay. You installed it on your phone, so that's where it is.

But these days WiFi is essentially leaking your position by virtue of the fact that we have databases, just comprehensive databases of network names and BSSIDs. And software is able through these messages to know what you're connected to and what device you are and so produce long-term tracking. And of course it's able to send the information out wherever it wants to. So we may be talking more about this in the future.

JASON: It's kind of disappointing when they stop at the current version with nothing prior. This Pie is on so few devices that it doesn't even make Android's developer platform or the dashboard for showing how many percentages all of the reporting devices of what OS - like Pie's not even on there yet, which basically means it's less than .1% distribution at this point in time, at least at the time that this report was pulled. So it's not even appearing on their report that tracks this stuff, let alone those are the only devices that actually get any of this protection.

Steve: So it's not inaccurate to say kind of at this point virtually all Android devices are currently enabling this broadcasting. And any apps which will, I imagine, will start appearing are able to - well, because it's meant to be system global broadcasts. So the fact that an app is saying I want to watch these doesn't, I mean, it's not going to make the app stand out. It's not something where Google can say, oh, you shouldn't be asking for this. No. It's a broadcast.

JASON: Yeah. And Google obviously knows this is not a good thing. They've patched it.

Steve: That's why they fixed it.

JASON: That's a bummer.

Steve: So, okay. Instagram was victim to some high-profile attacks. And this, I think, is really interesting because they have formally switched from SMS-based text messages to confirm your identity as an additional factor to, yes, third-party authenticator apps, meaning that they've switched to the time-based one-time passwords. So yay. What's interesting is that these high-profile attacks on Instagram users occurred even though they were protected by second-factor SMS-based multifactor authentication, which strongly suggests that the weakness of texting six-digit codes is not just theoretical, it's now actual. It's now proven. So anyway, it's good that they're doing that.

They have also moved to combat what they've termed "influence campaigns." They're showing, first of all, they are making available, the same way Twitter does, verified accounts. You need to prove your identity to them, your real-world physical identity, by for example sending them a picture of a government-issued ID proving who you are in order to become a verified Instagram account. And they've added the ability to view account information, which shows the date the account was created, the country the account is centered in, all ads that the account is currently running, any former

usernames that the account has gone under, as well as any other public accounts which share with it a large common set of followers. So bravo to Instagram. It's a little disheartening that it's taken them this long to implement these measures. But it really feels like, yes, we are getting there. And it's better than not getting there.

And I think what we're doing is we're, as a consequence of really pretty much what's gone on in 2018, and probably I guess in reaction ultimately to all of the press that the involvement of foreign parties or alleged involvement of foreign parties in the 2016 presidential election two years ago had, that we're now sort of setting a new standard for the way this can be done. And it really feels also as though SMS text messages, I mean, they were used because they were the lowest common denominator. Everybody who wanted to get authenticated who had a smartphone was able to receive a text message. You didn't need to install an app to do that. And so everyone's nontechnical relative was able to say, oh, I just got a six-digit code, I need to put that in. It's like, okay. That's easily done. Unfortunately, it wasn't secure enough.

So we've moved as an industry, I think, past that now. And I imagine it won't be long before authenticator apps won't be third-party authenticator apps. They'll just be part of the underlying OS. It's like, clearly it's time for that to happen.

JASON: That would be really nice, actually. I hadn't considered that as a solution. But as you were kind of talking about that, it's apparent to me that SMS is infinitely easier for the broad spectrum of people to do because their phone already does SMS versus having the app installed. That's one extra hurdle. People might not understand why they need to install this app to do it when it already works on SMS. Build it into the OS, that's a really great solution. That makes a lot of sense.

Steve: Yeah. And in fact I saw something, and I didn't take the time to track it down, but it looked like there was some provision for automatically registering the secret that drives the one-time password token within an interapp API. So if it were built in, and if there was a secure means for doing that, it might be possible for you to say I want two-factor authentication, and for the site to right then auto register your secret with the built-in authentication app. And so you're not even having to manually move that from one place to another. Which again, I mean, then that really makes it much more easy to use.

JASON: Well, and you even see that right now on SMS; right? At least I've noticed it before where SMS was the only choice for authenticating. That message comes through, and it automatically populates that into the app. The app that I'm using, trying to get into, recognizes that came from - I guess it came from the number it expected to get it, plugs the number in, you don't have to do anything. It's just like, great, we got it. Move on.

Steve: Yup. Unfortunately, the bad guys got it, too.

JASON: Yeah, right. Fair enough. Well, Google also has their other authentication where, when I log into their account, I will get just a popup on my phone that says is that you logging in, yes or no, so instead of doing a code or whatever. Where does that fall on kind of the security spectrum, as far as that's concerned? Better than using SMS, I assume.

Steve: It is, but it's proprietary. And so the advantage of the generic one-time password is that you get to choose which app you want. You're able to store them yourselves. There's a tool on iOS called OTP Auth, which I have started to use and like. I think I was using LastPass's for a while. And Leo and I talked about this. OTP Auth does use iOS Cloud in order to synchronize, which provides additional convenience. So that as long as

you set it up that way, if you add another account to it on one device, all your other iOS devices automatically know about it. So it ends up further improving the experience.

JASON: All right. So we talked about Instagram. I'm happy about that. I'm buttoning up my Instagram account, hopefully once that update comes through. It hasn't come through yet. Tell us a little bit about this OpenSSH disclosure vulnerability.

Steve: So we talked about a different one last week. In looking at some code associated with a recent security-related fix, a developer said, hey, wait a minute, there's another problem here. The problem was that there was a way that it was possible for someone to probe an SSH server to determine whether the username was valid or not. So separate from the password, it was regarded as an information disclosure vulnerability. And generally you want to only validate the username and the password together and say you got it or you didn't, rather than allowing someone to separately probe the username. So that happened, and we discussed it in detail last week.

Since then, the guys at Qualys found another somewhat related problem. But as a consequence of the way the OpenSSH developers responded to the first one, the Qualys guys were a bit put off and not really sure what they should do about it. So I want to share what Damien Miller, who is the OpenSSH dev, said with regard to the first problem, which we talked about last week.

He wrote: "Hi. Regarding CVE-2018-15473, a few people have asked why we just committed a fix for this without any secrecy or treating it as a security problem. The reason is that I and the other OpenSSH developers don't consider this class of bug a significant vulnerability. It's a partial disclosure of non-sensitive information."

He says: "We have and will continue to fix bugs like this when we are made aware of them and when the costs of doing so aren't too high. But we aren't going to get excited about them enough to apply for CVEs or do security releases to fix them. The following explains our reasoning."

He says: "First, this isn't 'user enumeration' [he has in quotes] because it doesn't yield the ability to enumerate or list accounts." He says: "It's an oracle" - that's the cryptographic technology term - "allowing an attacker to make brute-force guesses of account names and verify whether they exist on the target system. Each guess is moderately expensive, requiring one TCP connection and a cryptographic key exchange, limited in concurrency by SSHD's MaxStartups limit."

He says: "Second, very little else in the Unix ecosystem tries to prevent this style of information disclosure. Many network daemons will still happily return 'user not found' style messages; but, more importantly, system libraries are simply not designed to consider this a threat. They don't consider it a threat because usernames have long been considered the non-secret part of user identity, of limited use without actual authentication credentials."

He says: "In the absence of the underlying system stack being designed with this in mind, the best applications like SSHD can do is try to paper over the most obvious differences by avoiding behavior divergences in our own code and adding some prophylactic timing delays. But it's a losing battle."

And he goes on with some other details that I'll skip over because they don't really matter. Anyway, that gives you a sense for the philosophical position that he has. But he does say this. He says: "Finally" - and I think this is very salient. He says: "Finally, and perhaps most importantly, there's a fundamental tradeoff between attack surface and this class of bug. As a concrete example, fixing this one added about 150 lines of code to our pre-authentication attack surface." So that is to say this is the code that anyone

comes into contact to before they authenticate, meaning it is critical that they not introduce new vulnerabilities; that is, that there's nothing that they do wrong there.

So he says: "This one added about 150 lines of code to our preauthentication attack surface. In this case, we were willing to do this because we had confidence in the additional parsing, mostly because it's been reviewed several times, and we've conducted a decent amount of fuzzing on it. But given the choice between leaving a known account validity oracle or exposing something we don't trust, we'll choose the former every time." In other words, if it meant that fixing it might introduce a new vulnerability, they're just going to leave the oracle where it is, meaning that there's something probeable, technically.

So having read that, the guys at Qualys Security, in looking over that region of code, then spotted yet another somewhat similar problem. So there's a second similar oracle vulnerability. But due to the grumpy reaction of the previous similar issue, Qualys was somewhat put off. They wrote: "We understand that the OpenSSH developers do not want to treat such a username enumeration [and they have in parens] (or 'oracle') as a vulnerability." And of course, as we know, the OpenSSH developers even argue against the term "username enumeration." They say: "Although," Qualys says, "it is still quite useful in an attacker's toolbox." So they're asking: "But how should we coordinate this disclosure, then? OpenSSH developers, distros, please advise."

So anyway, our listeners know that I am constantly saying to Microsoft, would you please just leave it alone? Leave Windows be. Give it some hope of stabilizing someday. Don't keep messing with it. And this is exactly what the OpenSSH developers understand and are saying. I mean, OpenSSH is all about security. That's its sole benefit. You don't use it to play videogames. You use it for Secure Shell connections. So thank goodness they're being this security conscious. Last week we acknowledged that it wasn't such a big problem. They have certainly stated that clearly and for the record. And it's hard to argue with that. It's a little daunting that it was 150 lines of code required in order to fix this problem. I guess I'm glad they fixed it. It's not clear whether - I guess what they're saying is they will fix it if they can do so with confidence. But even so, they're not going to be issuing CVEs and considering this a big security problem, which I can certainly concur with. Interesting.

JASON: Yeah. So GDPR. I know that this is next up, and I've got lots of questions about this. I've got a few, anyway. This is interesting to me because I just assumed that the GDPR happening in the EU, all those rules were going to make sweeping changes everywhere. But apparently that's not happening.

Steve: Yeah. And so we'll explain what you're talking about and then talk about it because it is sort of an unexpected consequence. Nearly, get this, 1,200 U.S.-based news sites are deliberately remaining inaccessible, that is, they are blocking visitors from the EU as a consequence of the EU's adoption of the high-fine GDPR regulations, which has just freaked everyone out. And these deliberately blocked sites are not all obscure since they include, get this, the Los Angeles Times. Yup. Cannot bring up the L.A. Times from Europe. The Chicago Tribune. The New York Daily News. Dallas News. The Baltimore Sun. The Sun Chronicle. The St. Louis Post Dispatch. And Newsday. None of them are currently available to people in the EU. And as I mentioned, nearly 1,200 U.S.-based news sites, that leaves well more than a thousand smaller regional news sites which provide the bulk of news reporting overall and certainly are serving their communities.

So as we know, the European Union's GDPR regulations require websites to disclose their data collection practices in much more depth and detail than ever before, and also requires websites to obtain an explicit permission to collect this data from its visitors. The regulation also forces websites to provide a portal where users can see what data the website has collected about them and provide a way for users to delete this data. Now,

that's easy to say and easy to request. It's hard to do. I mean, it requires every single website on the Internet which can be visited by someone from the European Union to comply.

And so what have all of these nearly 1,200 sites done? They've said we don't need visitors from the EU. We'd rather block them than switch or invest in, right now, maybe forever, in abiding by these regulations imposed by another country on us. And except for these big sites, you could argue, like some Des Moines Dispatch or something, that no one in the EU probably wants to go there anyway. So blocking the ranges of IPs that are outside the U.S. or inside the EU, it's like, well, okay, you could argue they've saved themselves the exposure of being in breach of the GDPR, and it's not costing anybody any problem for them having to do that.

Remember that companies who do not adhere to the GDPR risk facing massive fines of as much as 4% of their annual revenue, which for major ongoing operations is significant. And again, I think it's rational for them to just say, you know, we didn't do this on purpose. Some other country has just decided that we're liable for behavior that nobody else in the world has a problem with. So, fine, we're just going to block you.

There is an interesting script and site monitoring this. Joseph O'Connor grabbed the domain verifiedjoseph.com, so it's data.verifiedjoseph.com which maintains a list on the fly of all the websites not available as a consequence of the EU GDPR. The script ran this morning because it was fresh. It shows up at the top the last time it was run. And as of this morning, when I put the notes together, there were 1,149 unavailable websites that he is monitoring and 147 that are. So more than a thousand. Actually, 1,002 more unavailable than are available. And some big names among them. So an interesting consequence, a side effect of one nation saying, one group of countries saying we're going to fine you unless you do this because our visitors might go to your website. It's like, oh, okay, fine. Block them.

JASON: Well, they're missing out on one way that they could make extra money because the Washington Post, apparently, back in May, their response was to put up an additional paywall targeted at the EU, a premium EU paywall that would remove ads...

Steve: Interesting.

JASON: ...and make extra off of that. So it's like, yeah, sure you can get it in the EU. We'll remove all the ads for you. You've just got to pay us more than a basic subscription fee in order to do it.

Steve: Actually, that makes a lot of sense.

JASON: They can make more money.

Steve: Yeah. There was, as a consequence of the legislation, this is a related matter, I think it was a 22% drop in cookies on websites just because of the GDPR. There were too many violations; and sites said, okay, we're not going to host third parties that are in violation of the GDPR because we can no longer afford to do that. So as an indirect consequence - because as we know third-party cookies are coming from ads and other unrelated sources typically. Those are down by about 22%. So that's been nice.

Speaking of cookies, four years ago, in 2014, Check Point's Malware and Vulnerability Research Group made a discovery. When a client connects to a web server, they ask for some resource from the server. That's what the URL is. And images and the text and all the stuff that the site has are requested. With those requests go any cookies that the browser has previously received from that domain as a consequence of previous queries.

So as we know, cookies are the way, they're a mechanism that Netscape added in the early days of the web to allow this notion of logging into a website.

The actual act of obtaining a web page is stateless. The browser says give me a bunch of stuff, and it gets it. And then, if you click on a link within that site to another page, the browser asks for that page, and all of that page's stuff. But nothing links you and those two pages together. There is no tracking in the original web. That was created by Netscape where, in response to the first request, which would not have had a cookie from your browser because your browser has never been there before, the web server goes, oh, let's give him a cookie.

And so a nonce, a one-time - a nonce is a number once. Just a random, pseudorandom blob of noise is handed back to the browser in the first reply. And now the browser has a cookie, and so it sends it back on all subsequent requests, not only for the duration of the other things on that page, but when you click a link, it sends it back with the link you're clicking. And as the browser then obtains the next page and all of its assets, that cookie keeps going back. So that creates a stateful connection to the site.

Okay. So it turns out, not surprisingly - so remember, in a query the browser is returning the cookie it received. Turns out that what Check Point discovered was in a huge number of exposed residential routers, so-called SOHO, Small Office Home Office routers, the web server that was publicly exposed was not correctly checking the length of the cookie it received. It assumed that the browser was sending back the cookie it had been given. But you could have malicious cookies, or in this case Misfortune Cookies. Thus this name.

So back four years ago, researchers from Check Point's Malware and Vulnerability Research Group uncovered this critical vulnerability which was at the time present on millions of residential gateway devices across models and makers. It got assigned at the time a CVE number, 2014-9222. It was a severe vulnerability, allowing an attacker to remotely take over the device, obtaining administrative privileges. At the time, they detected approximately 12 million readily exploitable unique devices on the Internet across 189 countries, making it one of the most widespread vulnerabilities that had been seen at the time. And research suggested that the true number of affected devices that were not detectable might have even been greater.

So anyway, the point is that anyone making a query to the website could, along with the query, essentially inject a malicious cookie into the query which would induce a buffer overflow and allow that cookie to contain code which the web server would then execute, making this an incredible potent remote code execution vulnerability.

So now move forward four years. It's 2018. The Industrial Control Systems Cyber Emergency Response Team - there's a mouthful, that's the ICS-CERT - has identified this same vulnerability, currently widespread in medical device systems. There's a very commonly used piece of equipment or technology known as the DataCaptor Terminal Server (DTS), which is a medical device gateway developed by Qualcomm Life subsidiary Capsule Technologies SAS, which is widely present in medical management systems. The gateway is used in hospitals to connect medical devices into the larger network infrastructure.

The cybersecurity firm CyberMDX discovered the presence of the flaw, which can be exploited by attackers to conduct remote arbitrary memory writes, just like four years ago, which could lead to unauthorized login and code execution. The vulnerability in the device is present in a software component called the RomPager from Allegrosoft, which is used by the DTS web interface. And according to CyberMDX, the version of RomPager in use is an older version, earlier than 4.07, where this problem was fixed, and the older version is susceptible to the Misfortune Cookie attack. More up-to-date versions are not affected because they were patched.

Turns out that even more worrisome is the fact, given where these things are being deployed, that the vulnerable version has remained in use through the last four years only because the vendor would need to pay to receive updated firmware and has elected not to do so. The only silver lining in this cloud is that the web server component, which is where the vulnerability is, is only utilized and required during the initial configuration of the device. So the embedded vulnerable web server can be disabled once the device is set up and configured. On the other hand, we all know the likelihood of that happening in the real world. It's almost not going to. Maybe a few responsible admins who become aware of this, who are listening to this podcast, will say, oh, that affects me, and they'll turn this off. That would be great.

We've often talked about how devices like this which qualify as Internet of Things devices have to take responsibility for keeping themselves up to date. These things have to periodically check with the mothership to see if there's an update for them and fix themselves autonomously. We've moved away from SMS text messages for second-factor authentication to time-based. We have to adopt similarly autonomous updating of IoT stuff. Our routers have to do it. Our light bulbs have to do it. If it's on the Internet, if it's accessible, it's got to update itself. Again, we're in the early days yet. And we know that because we see these things changing as we slowly and reluctantly, kicking and screaming, mature. But, boy, it's not happening quickly.

JASON: Yeah. I'm filled with a little bit of doubt, but I'm at least enlightened by the fact that you say that it's early days and that it worked for SMS, it could work for this, too. But, man, it's an uphill battle, I think, with IoT.

Steve: It is a big ask. And you are right, Jason, to be filled with plenty of doubt. So Magneto is an Adobe company that boasts it's powering twice as many top retailers as any other provider. What they are is an eCommerce provider. And, for example, Coca-Cola and Burger King and many other users leverage the eCommerce platform which Magneto offers.

JASON: Is it Magneto or Magento?

Steve: Ah, you're right. Well, M-A-G-N-E-T-O. I don't know how you pronounce it.

JASON: No, if you click, yeah, I think if you click through it's Magento.

Steve: Oh, you're right. It's M-A-G-E-N-T-O. Yeah, yeah, yeah.

JASON: I mean, I think it's a Marvel Comics character, and he's pretty awesome, but Adobe is even awesomer. I don't know. No, I don't think that works. Yeah, I think it's Magento.

Steve: That's interesting because throughout the notes I have Magneto Core. Magento. Anyway, Magento, sorry.

JASON: It's all good.

Steve: Magento, looks like Magento. Good, I'm glad you caught that, Jason.

JASON: Yeah, no worries.

Steve: Over the past six months Dutch security researcher, whom we've often referred to on the podcast, Willem de Groot has found a malicious credit card skimming script which, now, I've got in here Magneto Core, but it's magentocore.net. So sure enough, I'm checking the actual URL here. So, yes, which he named Magento Core. And get this.

He has found this on 7,339 Magento-hosted storefronts. So that's just the places where end users go and say, oh, I want to buy something, and put in all of their name, their address, their whatever information it asks for, email, phone number, and credit card information. On 7,339 storefronts there has been, and in - get this - 5,172 domains right now today there still is, skimmer script running which grabs all of that information and sends it back to bad guys, apparently in Russia.

He explains on his posting of this that victims of Magento Core who use the storefronts have their credit card and identities stolen. He writes that the group that is behind this has not slowed down; that new brands, that is, new storefronts, are hijacked at a pace of 50 to 60 stores per day over the last two weeks because he's got now a scanner running, looking for this illegal scripting. And apparently you can just do a worldwide web search for the appropriate pieces of the script, and Google will find it for you.

The Magento Core skimmers gain illicit access to the control panel of an eCommerce site, often through brute-force techniques, automatically trying lots of passwords, sometimes for months. So nothing is preventing them from doing long-term brute-force cracking of the login to the eCommerce site control panel, which suggests that, unless not in use, that control panel should be taken offline. Unless you actively need to get to it from the web, turn off the control panel. That's clearly the crux of the vulnerability here.

Once they succeed in gaining access, they embed one line of JavaScript into the HTML template. And it's just a standard JavaScript invocation to https://magentocore.net/mage/mage.js. That script gets added to the pages and sucks off everything that the user does. It records keystrokes from unsuspecting customers and sends everything, he writes, in real-time to the magentocore.net server registered in Moscow. Also the malware includes a recovery mechanism. In the case of the Magento software, it adds a backdoor to the cron.php file which will periodically run to download malicious code and make sure that it comes back after it gets removed. And there's even a clean.json backup, which is PHP code, which removes any competing malware from the site.

So, wow. Again, this feels to me like something that Magento needs to address by doing something like removing long-term global access to the control panel by default. So like requiring internal access only. Or if you make it available on the public interface, then have it automatically shut down after some minutes of non-use. Not just log you out, but become unavailable so that some other action is required in order to bring it back up publicly. This is a big problem that apparently, you know, this has been going on for six months. Certainly they're aware of it, and apparently nobody's doing anything about it. Seems wrong.

Okay. So our Trusted Platform Module, the TPM, is present now - I remember, what, 10 years ago, because it's been around for a long time, it wasn't clear whether you were going to have it or not. My previous motherboard, which was about 15 years old, had a socket where - it was a, can't remember now, a gigabyte board, I think. It had a socket where you could plug in an optional TPM module, but it wasn't just built into the motherboard. They are now. You can verify that just by going to your BIOS. Go the security tab in the BIOS, and you will see you can turn it on or off. You can initialize it and so forth. So pretty much everybody's got TPM onboard.

There's a problem. And it's not with some random manufacturer's backdoor or bad implementation. It's actually in the spec. Researchers with the South Korean National Security Research Institute identified a flaw in the Trusted Platform Module spec relating to the way power modes are handled. They presented their findings, again, at that recent USENIX conference. ACPI is the longstanding Advanced Configuration and Power Interface spec and API which defines the power states for a system and the hardware registers for supporting power management.

There are global states known as G0 through 3, which are respectively working, sleeping, soft-off, and mechanical-off. And then there are local per-device states which are the ones we're more familiar with. There's S0 and S1, which are working and power on suspended. There's S2, which is the same as S1, that is, power on suspended, but the CPU is also powered off. There's S3 which is the well-known sleep, where all devices are powered down except RAM. It keeps RAM alive. And then S4 is the formal name for hibernation, where the RAM is written out to static storage, and then it, too, is powered off. So sometimes you see those in a BIOS. And essentially what ACPI does is it provides per-device power management. So in order on a system that's got all kinds of stuff all over it, it's possible to power down selectively different pieces of the entire system in order to get overall improved power management.

Well, it turns out there's a glitch in the way the TPM interfaces with ACPI such that it's possible for the TPM's boot-time hash validation values to be intercepted and replaced. So the TPM is used when we boot our systems to implement so-called "secure boot." The idea with secure boot is you start with something you absolutely trust. You have to have a trust anchor, and that's the TPM. It's in hardware. You cannot read stuff out of it. You can only ask it to do work for you.

It is like a Secure Enclave on our PC-based systems. It verifies the signature of the first thing that is before it's run. Basically it oversees the entire boot-up sequence where, stage-by-stage, the signature is verified of the software by taking its hash and checking it against the valid hashes that the TPM has, the idea being that, if you start from absolute secure, and you load and check each module in turn, there's no way for bad stuff to get in until you finally load the OS and make sure it hasn't been tampered with and then turn over control to it.

Well, turns out by being tricky with power states these researchers have found a way to subvert that secure chain boot-up process, and they demonstrate it. They have a proof of concept, and it works. So ultimately what we're going to look at is another round of firmware updates. In this case it's not the firmware for the processor, it's the firmware for the TPM. It is a standalone microcontroller device that has firmware that runs instructions, and they need to be updated to fix this. If you're in a high-threat environment, there is a short-term mitigation which anyone can employ if your BIOS allows you to disable the system's S3 sleeping state. And many BIOSes do that.

Historically there have been systems that did not awaken from S3, from sleep, correctly. And so an option in the BIOSes for years has been to disable S3. The hack these guys have come up with absolutely requires that S3 be enabled. And many people don't use S3. They just don't have an occasion to sleep their systems. Certainly enterprise servers, probably lots of enterprise desktops don't. Mostly laptops. But remember that when you sleep your laptop, there's still a power drain because RAM is being kept alive.

So if you hibernate your laptop when you close the lid, or it does a full shutdown, even a laptop may not be needing the S3 state. So you can probably disable it without any inconvenience to you, and in doing so you are protecting yourself until Intel, probably, gets around - or I guess Infineon is the actual maker of most of these TPM chips. So maybe it'll be from Infineon, then maybe Microsoft will push this, or Intel will. We'll see how we get this. But essentially the entire industry has been put on notice that it is not safe to trust TPM at the moment. It can be subverted.

JASON: What else have we got here?

Steve: Well, so I've mentioned throughout the podcast already many CVE numbers, which is the database that we've really become quite dependent upon, and not just in the U.S., but globally. In the show notes I have a chart courtesy of Bleeping Computer, who covered this story, showing the black bars as the funding from 2012 through 2016 versus

the white striped bars are the number of CVEs published. And so for our listeners who are only getting audio, the funding was in 2012 and moving forward each year: 6.7 million, then dropped to 4.8, then dropped to 2.8, then dropped to 1.7 in 2015, to rise to 4.0 in 2016. But still, in 2012 it was at 6.7. Across that same period of time, the number of CVEs published has about doubled. In 2012 it was 7,370; and in 2016, 14,472 reported vulnerabilities.

So I'll share what Bleeping Computer had to say since they summarized this nicely. They said, and I've got the link to the Bleeping Computer story for anyone who wants it: The CVE, they said, "was created in 1999 by the MITRE Corporation using U.S. government funding. It's a database that contains identifiers - tracking numbers - for security vulnerabilities. Since its creation, the CVE system has been adopted by the public and private sectors. Most modern cybersecurity software use CVE numbers to identify and track cyberattacks exploiting particular software bugs. Despite being a U.S. creation, the system has been widely adopted in countries all over the globe, which use and recognize the CVE identifiers issued by MITRE's staff and industry partners.

"But in recent years," writes Bleeping Computer, "the CVE system has been under stress. Its problems became evident in late 2015 and early 2016 when a large number of security researchers reported long delays in receiving CVE numbers for the vulnerabilities they were reporting. At one point, a few of them united to create an alternative vulnerabilities database known as the Distributed Weakness Filing (DWF).

"At the time, MITRE said the CVE number assignment delays were caused by the increased number of software vendors compared to the late '90s and early 2000s, but also because of the proliferation of software-driven SCADA industrial control equipment and Internet of Things devices." In other words, more companies creating equipment, more security researchers reporting, and more things going wrong to report.

"Both factors," writes Bleeping Computer, "contributed to a huge rise in vulnerability reports, with which the CVE staff wasn't managing to keep up to date." And of course at the same time their budget was being cut every year. "A late 2016 report found that MITRE's CVE failed to assign CVE numbers to over 6,000 vulnerabilities discovered in 2015." So there were 14,000 numbers assigned; 6,000 weren't. So actually more than 20,000 vulnerabilities.

"So in March of last year the U.S. Senate got involved to investigate the problems and concluded: 'From 2012 to 2015, the program has received on average 37% less year-over-year funding.'" Also, they said: "'The documentation produced by DHS and MITRE shows that the CVE contract vehicle is both unstable and prone to acute fluctuations in scheduling and funding.'

"To solve this issue, the Senate's Committee proposed that DHS officials move CVE's funding from a contract-based funding scheme into the Department of Homeland Security itself, as what's known as a PPA (Program, Project, or Activity) funding line item. The Committee believes this would provide a constant stream of funding, reducing huge budget fluctuations, and keep MITRE focused on running the CVE database instead of always worrying about its future funds."

So that all sounds good to me. I hope that that happens and that they're given enough money because it's very clear that this is providing a vital clearinghouse database for managing all of these cybersecurity threats, which, as we know, can be significant.

Mozilla has announced a very welcome change of anti-tracking approach for future Firefoxes, which is available in the Firefox Nightly builds for anyone who wants to enable it. In their posting they said: "Anyone who isn't an expert on the Internet would be hard-

pressed to explain how tracking on the Internet actually works." And of course we've been left breathless on this podcast doing just that.

"Some of the negative effects," they write, "of unchecked tracking are easy to notice, namely eerily specific targeted advertising and a loss of performance on the web. However, many of the harms of unchecked data collection are completely opaque to users and experts alike, only to be revealed piecemeal by major data breaches when they occur. In the near future Firefox will, by default" - and that's the key, you don't have to go anywhere and flip anything on - "by default, protect users by blocking tracking while also offering a clear set of controls to give our users more choice over what information they share with sites.

"Over the next few months," they write, "we plan to release a series of features that will put this new approach into practice through three key initiatives." Get this. Under improved page load performance they wrote: "Tracking slows down the web. In a study by Ghostery, 55.4%" - okay, 55.4, more than half - "of the total load time required to load an average website was spent loading third-party trackers." Right? Third-party trackers are more than doubling the amount of time required to load our pages.

They say: "For users on slower networks" - I would just say for anybody - "the effects can be even worse. Long page loads are detrimental to every user's experience on the web. For that reason, we've added a new feature in Firefox Nightly that blocks trackers that slow down page loads. We will be testing this feature using a shield study in September. If we find that our approach performs well, we will start blocking slow-loading trackers by default in Firefox 63."

So that's very cool. What that means is they will instrument Firefox - five seconds is what I saw referenced elsewhere. So if a tracker takes more than five seconds to get itself loaded, it will be blacklisted, and it will no longer be loaded by Firefox. To which I say bravo. And it's interesting, too, because think about it. Trackers don't have any particular incentive to be speedy. Websites do. But websites don't have control over the trackers that their ads load or that other third-party assets load. So websites are being hurt indirectly by the trackers that the assets they're invoking are loading. So this really is great. This says to trackers, get your servers sped up so that you are able to respond to the requests you are creating on web pages; otherwise, be blocked. So I think this is great.

JASON: Fantastic news. And you have to imagine that Firefox, the work on Firefox Focus, which came out I think in 2015...

Steve: Ah, right.

JASON: Initially for iOS. Now it's on Android. And the focus, no pun intended, when they first launched was to basically block the tracking. And then it's kind of expanded into other privacy kind of emphasized sort of uses. But, yeah, that's great. Any way that it can speed things up and clean up the experience, just remove the muck.

Steve: And Google is in, of course, as we know, a bit of a dicey problem because their revenue all comes essentially from web-based advertising which is enhanced by tracking.

JASON: Sure.

Steve: So it'll be interesting to see what they do, if they follow suit. Also, that was the first of three. Second of the three is removing cross-site tracking. Mozilla writes: "In the physical world, users would not expect hundreds of vendors to follow them from store to store, spying on the products they look at or purchase. Users have the same

expectations of privacy on the web; and yet, in reality, they are tracked wherever they go. Most web browsers fail to help users get the level of privacy they expect and deserve.

"In order to help give users the private web browsing experience they expect and deserve, Firefox will strip cookies and block storage access from third-party tracking content. We've already made this available for our Firefox Nightly users to try, and will be running a shield study to test the experience with some of our beta testers in September. We aim to bring this protection to all users in Firefox 65 and will continue to refine our approach to provide the strongest possible protection while preserving a smooth user experience." So that's big. That's not the DNT, you know, please don't track me flag that you can set. This is the browser itself not honoring third-party, that is, cross-origin content. And again I say yay.

And then the third of three, mitigating harmful practices. They write: "Deceptive practices that invisibly collect identifiable user information or degrade user experience are becoming more common. For example, some trackers fingerprint users, a technique that allows them to invisibly identify users by their device properties, and which users are unable to control. Other sites have deployed cryptomining scripts that silently mine cryptocurrencies on the user's device. Practices like these make the web a more hostile place to be. Future versions of Firefox will block these practices by default."

And they say: "Note that these features are currently available in the Firefox Nightly builds. Users wishing to experiment with them may manually enable them." And in the link that I have at the top of the story to Mozilla's blog, there are the details spelled out for how to do the Firefox Nightly enablement of these if you're curious. And again, Firefox is still my go-to browser, mostly because it handles side-located tabs so nicely, and I desperately need tabs, horizontal tabs running vertically down the side of my browser window. So I'm sticking with it for now. And I'm sure happy with these privacy-related features that they're supporting.

JASON: Yeah, very nice.

Steve: I'll just note that Microsoft did release a non-Patch Tuesday substantial update which, fortunately, my Windows 10 machine that I'm talking to everybody and you over, Jason, was able to digest in time for the podcast.

JASON: Right.

Steve: It wasn't security related. It was a whole host of non-security-related problems. I won't enumerate them because literally there are too many. But reading the list as I did of the things that they fixed makes you glad that Windows 10 was working beforehand because, wow, I guess there's all kinds of little fringe edge conditions and things that they are working on. And as I've said, if they would only just leave it alone, then they could fix these things and not be breaking more things at the same time. But that's clearly falling on deaf ears because here we are at Windows 10, even though earlier Windows worked just fine. Anyway, grumble, grumble.

I found a really nice note from a Rick Zich - he actually helped me pronounce his name, it's spelled Z-I-C-H but pronounced "Zeke" - in Tucson. He said: "Steve, I built this machine for my father's business, and it had been running good for a few years. Over those years we replaced the motherboard and also the power supply and added a high-end graphics card as he now pushed the computer to a large 50-inch" - wow - "4K TV screen and wanted the best picture he could get.

"Recently he started to get some pink screens of death." And Rick says, "Yes, a pink screen, not a blue screen. Evidently, when you are running Nvidia 1070 or other high-end video cards, the blue will change colors to pink. As strange as I found that, the point

was that the machine was rebooting randomly. I also tested that concept. I took out the video card and did indeed continue to get BLUE screens with the card out. I could not for the life of me figure it out. Lots of help on the Internet for things like make sure to update the Intel video drive even though it's not active, run your Windows updates, or reset your RAM." I guess that means pulling the DIMMs out and pushing them back in again, which I certainly know well. That's kept a few of my machines going.

He said: "I had individually tested all the pieces of hardware in a separate machine, and all would run fine with the other machine. He was running a strange version of Windows 8.1, so I even upgraded him to 10 and even put it on a 500GB SSD, which was brand new. However, I had cloned the disk over to the new one and then ran the Windows upgrade. I finally thought, could I have cloned over corrupt data? So I pulled out my SpinRite on my bootable USB stick and ran it on Level 2. I had no expectations of it working, as I thought SpinRite would only fix damaged disks and not damaged data. But what did I have to lose? Well, it's been a week, and no reboots yet from pink screens. Yeah!!!" he says with three exclamation points.

And I'll just note that it is true that SpinRite's focus is on recovering sectors which the drive says it cannot read. But there is, as we've often talked about, a disturbingly large gray area. And as defects get large, it is possible for the drive to miscorrect those sectors so that it believes it has performed an effective error correction, because it's actually statistics, when in fact it has not. It's a little bit like a parity error which can correct even or odd parity. And so it will detect a single bit flipping because that will change the parity of the entire block. But two bits flipping it won't detect because that maintains the same parity. ECC is the same way. SpinRite won't be fooled.

So what happened was SpinRite realized there were sectors which were being miscorrected by the original hard drive. It fixed them correctly and rewrote them so the drive then worked properly. So it's always surprising people how much there is under the covers of SpinRite that is not revealed by the fact that you just run it, and it fixes everything. But that's what it does.

JASON: Does it ever surprise you at what it's capable? Like at this point you've seen it all.

Steve: Oh, yes. Oh, yeah, yeah. And in fact I've talked about how my bookkeeper operations gal, I had set her up with a mirrored RAID, and the first drive died. And she didn't want to bother me. And I've since explained to her, "Sue, that's when you bother me. That's why we have a RAID, to be redundant." And in fact I've talked about this on the podcast before. It annoyed me that the RAID continued to say, I'm damaged, but okay. A consumer RAID should say, "Okay, one drive is down. Fix me now while you can still clone the good drive to another good drive." But it didn't. It just kept on going, for months, until the second one died. And that's when I got the call from Sue saying, "My computer won't boot." And I said, really. So I made a house call. And I said, "Wow, both drives are dead." And that's when she confessed that one had died a long time ago. And I just said, "Oh, okay. Next time, when the first one goes, believe me, you're not inconveniencing me, you're saving me time."

JASON: Right.

Steve: So the point is the system was completely down. I ran SpinRite on it. I don't remember which of the two drives came back. One of them did. Maybe they both did, I don't remember. And then I was able to - there was zero data loss, and we brought the system back up, and I think I probably put new drives on it because why not? And she's off again and hasn't had any problems since. Although she is complaining that it's getting a little old. So it's time to get her fixed up. But anyway, SpinRite does surprise me, often.

JASON: Awesome.

Steve: Okay. SonarScoop.

JASON: Snoop, Snoop.

Steve: Oh, Snoop, yes. Why do I do that? Yeah, you and I both did that.

JASON: No, I totally did it before the show. And I was, what is a sooner - I think it corrected to SoonerScoop, for the Sooners. It was a sports site for the Oklahoma Sooners football team. And I was like, this is not right.

Steve: So I'll tell you, this is the best time of year for the podcast because we've got the USENIX conference, and we've got Black Hat and DEF CON. I mean, it is just security conference heaven. So this is another piece of research that was shown at the three weeks ago now Baltimore USENIX security conference. In the show notes I have a picture from their research PDF which they published as part of the conference proceedings. It shows the back of a Samsung Galaxy S4 with the microphone at the top of the phone and the speaker on the back. Oh, I'm sorry, the microphone at the bottom that you normally talk to and a speaker at the bottom. And then on the other picture is a top-located microphone and a top-located speaker. So the top one is where your ear normally is, and obviously the bottom mic is where your mouth normally is. But to increase noise cancellation and the utility of the phone, they put both the microphone and a speaker also on the opposite ends. So you've got a top microphone, a top speaker, a bottom microphone, and a bottom speaker.

Well, the next page of the show notes I show their simplified diagram of what that means. That means you can ping from the top speaker an ultrasonic sound that the user will not hear, which will be received after some length of time based on how far away the person is and the rate at which sound travels, will be heard by the upper microphone. Similarly, you can ping from the bottom speaker, and it will bounce off the underside of the user's finger and back into the bottom microphone. In other words, classic sonar.

And, now, to triangulate you really would want the sensors to be mounted on adjacent corners of the area. That way, if you take the two distances, it's called "triangulate" because the two sensors lie on one edge of a triangle, and then the distance that they're each sensing informs it of the length of the other two lengths of the triangle, allowing it to uniquely determine the apex of the triangle not containing the sensors, thus triangulation. We don't have that here. We've got kind of arbitrarily located sensors. So it's less than ideal. But their research demonstrates that they are able to, not perfectly, but to significantly determine the instantaneous position of a user's finger which is in contact with the screen.

So reading from their abstract at the top of their research, they said: "We report the first active acoustic side-channel attack. Speakers are used to emit human-inaudible acoustic signals" - you know, like bats - "and the echo is recorded via microphones, turning the acoustic system into a smartphone sonar system. The echo signal can be used to profile user interaction with the device. For example, a victim's finger movements can be inferred to steal Android unlock patterns. In our empirical study, the number of candidate unlock patterns that an attacker must try to authenticate herself to a Samsung S4 phone can be reduced by up to 70% using this novel acoustic side-channel." So, note, not reduced to zero, but reduced by more than half.

"The attack is entirely unnoticeable to victims. Our approach," they write, "can be easily applied to other application scenarios and device types. Overall, our work highlights a new family of security threats." So in terms of details, they used a dictionary of 12 unlock patterns. Okay, so not the universe of possible unlock patterns. They just, you

know, they were looking for does this kind of work at all. So they used a dictionary of 12 unlock patterns in their tests which contained 15 unique strokes.

The data collected from 12 volunteers was fed into an AI learning machine model for classification of each stroke; and, as expected, the classification accuracy was significantly higher when simultaneous input from both microphones was considered. The researchers reduced the average number of correct candidates from the 12 unlock patterns to 3.6. That's average. In some instances, the analysis eliminated all guesses and revealed the single correct pattern uniquely.

I looked at the research, and I won't go into any more detail here because everybody gets it. But I have a strong gut sense that we're going to see this evolve. It uses sensors available in our devices. It could certainly be used for benign purpose, like moving your hand around in front of an app in order to do something, and having the app respond. It's not super high fidelity. Again, the sensors are not located exactly where you want them. Was it the Apple, I mean, it was the Amazon phone. They had like some funky sensor in each corner of the screen for a while; didn't they, Jason?

JASON: Yeah. And that was meant to kind of change the parallax of what you were looking at. It was meant to kind of, like, from the four points be able to determine your perspective and shift accordingly.

Steve: So to see your face so that they could change the screen in order to make it look more depth-y.

JASON: It was a lot of hardware for a very minimal application purpose.

Steve: Yeah.

JASON: But, no, what this reminds me of is, if you remember, I think it was like three or four years ago at Google I/O, Google had shown off something called Project Soli, which was also...

Steve: Oh, yes, yes, yes. And I am so jazzed by that, where you could do, like, push buttons and turn knobs and things. Oh, I really want to see that happen.

JASON: I haven't really heard much about it since then. And I think their idea was that it would possibly be integrated into wearables, like smartwatches of some sort, so that you wouldn't have to touch it, but you could still do the controls. And then there's also something else that I read about called, what was it, it was called FingerIO, which there's a demonstration. And that seems a little bit more aligned with what SonarSnoop is all about.

Steve: Yeah. And Soli was EM radiation. So it was very low-level electromagnetic radiation that gave it extremely high resolution in return for having to have a lot of processing behind it.

JASON: Right.

Steve: But anyway, I thought this was very cool, very clever, something no one had thought of before. And I wouldn't be surprised if we see this evolving in the future. This feels like something that could be taken from this initial research out to its next level. So props to these guys, and a nice piece of research.

JASON: Yeah.

Steve: No matter what you call it.

JASON: I'm curious on a device-specific perspective because, like you were saying, every phone has a microphone and a speaker in a different location. And so for this to work on a broad spectrum of Android devices, I imagine those locations create different hurdles or different requirements for where exactly on the screen your finger is at any given moment. If the speakers and microphones are over here versus over there, does that then narrow down the accuracy differently?

Steve: Well, and so, for example, one thing that's - if somebody wanted to be clever, they could create a game which inherently requires you to move your finger around and touch different areas of the screen.

JASON: Oh, to learn, yeah.

Steve: While, yes, while you're playing the game it's doing sonar on you, and you don't know.

JASON: It's calibrating.

Steve: Yes.

JASON: Yup. Dang it, that's too wise. That's too smart. And you know what, that'll probably happen. But this only really, from a security standpoint it seems like this really only works if you're doing a swipe pattern; right? Would there be any indication that it would work with tapping in a PIN or something along those lines?

Steve: And, see, that's exactly what I mean when I say this feels like first gen.

JASON: Yeah.

Steve: And so they demonstrated the concept is viable. And so I wouldn't be surprised if, before long, they could turn this into read keystrokes on a keyboard.

JASON: Totally. Well, that's another thing this kind of reminded me of, the whole thread from years ago about people being able to listen to someone typing on a keyboard and being able to reconstruct what they're typing based on the sound of those keys clacking.

Steve: Yeah, yeah. That's just bizarre, but it's true.

JASON: Yeah. Oh, this stuff is crazy. Very interesting. Well, we'll certainly find out more about that as we go along. But have we missed anything?

Steve: We're done, baby.

JASON: We hit it. We hit the end. Our outro has been SonarSnoop.

Steve: Oh, my god, and look at that. My clock says one hour, 50 - oh, there it is, two hours. Two zero zero zero zero.

JASON: We made it. Steve, thank you so much for all the news and diving in deep on all this stuff. It's always awesome to get the chance to do this with you, and I get another two weeks, so I'm looking forward to that. For folks who don't already know - you already know, but as a reminder, GRC.com if you want to find all the cool stuff that Steve is up to. You can get SpinRite there, which you heard him talk about a little bit ago, the

best hard drive recovery and maintenance tool. Get your copy there. I'm sure information on SQRL, all sorts of stuff found at your site; right?

Steve: Yup, yup.

JASON: So make sure and do that, GRC.com. And if you want to find this show, I think you're also posting audio and video over there, as well; right?

Steve: We have audio, video, and the transcripts are uniquely available over at GRC.com/securitynow.

JASON: Yes. And you'll want those transcripts, as well. So check it out over there. You can also come to our site, of course. We have the show, every single episode cataloged at TWiT.tv/sn. You can find audio and video there. Subscribe to the podcast. Of course you can find this on YouTube and anywhere. You know, if you have a smart TV, just do a search in the console there, you're going to find it. And, yeah, I think that's about it. We record live every Tuesday, starting at 1:30 p.m. Pacific, that's 4:30 p.m. Eastern, 20:30 UTC. And TWiT.tv/live if you want to watch us live. Thanks to the folks in the chatroom. They were throwing out some valuable information to us, as well, that kind of made its way into the show. So that's what I love about the live experience. People get in there, they're chatting about it, and sometimes it kind of makes its way in. So we appreciate their participation, as well.

That is it. Steve, thank you so much, man. And we will see you next week.

Steve: Thanks, buddy. I'll talk to you next week.

JASON: Take care. Another episode of Security Now!. See you later.

Steve: Bye.

JASON: Bye, everybody.