SECURITY NOW!

Transcript of Episode #737

Biometric Mess

Description: This week we check in on the frenzy to turn CheckM8 into a consumer-friendly iOS jailbreak, on another instance of stealth steganography, on a number of changes to Firefox's URL display, and on the state of Microsoft's ElectionGuard open source voting system. We also look at a very serious flaw that was just found in Linux's Realtek WiFi driver and some welcome news from Yubico. We touch on a couple of miscellaneous media tidbits, then take a look at the ramifications of two recent biometric authentication failures and consider the challenges and inherent tradeoffs of biometric authentication.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-737.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-737-lq.mp3

SHOW TEASE: Coming up next on Security Now!, of course Steve Gibson is here, along with me, Jason Howell. Steve's going to dive into, not one, but two big biometric mishaps on two of the biggest Android smartphones out on the market today. Well, one of them's almost out on the market. Also security display changes in Firefox 70, a big flaw in Realtek WiFi drivers, YubiKey for local Windows login is released, and a whole lot more, next on Security Now!.

JASON HOWELL: This is Security Now! with Steve Gibson, Episode 737, recorded Tuesday, October 22nd, 2019: Biometric Mess.

It's time for Security Now! with Steve Gibson. I'm Jason Howell, filling in for Leo. Leo's gone for a month, four episodes back to back that I'm going to be sitting in this chair along with you, Steve Gibson. How are you, Steve?

Steve Gibson: Hey, Jason. Great to be with you.

JASON: Great to see you.

Steve: I was looking. I think it was like 726 or 23 or something was the last time we were together.

JASON: Something like that.

Steve: So it's been not that long.

JASON: Yeah.

Steve: And now we have four episodes...

JASON: Absolutely.

Steve: ...that we're going to do while Leo is off on his cruise through Greece or wherever he's going.

JASON: Yeah, you know, to be honest, like somebody asked me the other day, well, where is Leo going? And I was like, you know what? I can't remember. I lose track of where Leo's going. So I knew that he was going somewhere where Internet would be difficult. He made some comment about that. But good for him. He should. And, boy, we should all look forward to a month off somewhere down the line in our lives.

Steve: Well, we have Episode 737 for October 22nd. And there were a couple of things that hit the news that, I mean, I was aware that you were going to be my co-host. Of course you're Mr. Android, and these both affected Android devices.

JASON: Yes.

Steve: Well, and there are, like, big-time security issues for us because it's about biometrics. Two high-profile failures of biometric authentication were in the news in the last week. So I thought, that's perfect for us to talk about. And I've been wanting to talk about it in general. We've been talking and sort of like following biometrics for at least a decade. We were talking a long time ago about the fingerprint readers at the Disney theme parks and, like, wait a minute, do I want to be giving Disney my fingerprint for the privilege of, I don't know what, like getting in through the gate more easily than I would otherwise? And so even back then we made a joke of it. We were talking about giving them your knuckle instead of your fingerprint. It's like, don't give them your fingerprint. Give them your knuckle.

JASON: Were you saying to punch them? Is that - yeah, I'll give you my authentication.

Steve: So anyway, we are going to finish by talking about that. But we've got a bunch of, of course, as we always do, interesting news of the week. And I'm staying true to my promise not to beleaguer our listeners with talks of crypto malware because we sort of had, for a while there this summer, we were just - it was becoming the ransomware crypto podcast, and I think we maybe pushed our listeners a little bit too far.

But we're going to check back on the frenzy to turn the CheckM8 iOS jailbreak, well, iOS vulnerability, the Boot ROM vulnerability, into a consumer-friendly iOS jailbreak. That's underway. That was the title of last week's podcast was CheckM8 because it was such a big deal, the idea that all devices from the A5 chip up through the A11 all incorporate a Boot ROM flaw that Apple can never change because it's literally in ROM, which is soon to be leveraged into a jailbreak. We're not quite there yet, but we'll talk about that.

We also have a new instance of stealth steganography which is sort of interesting because it's the whole concept of hiding something in plain sight. We've got a number of forthcoming changes to Firefox's URL display in the next version of Firefox, which will be number 70. Also an update on the state of Microsoft's ElectionGuard open source voting system. Also there is a - of super important interest to all Linux users, and apparently that includes again Android devices, any Linux or Android device using the Realtek WiFi chip - a buffer overrun has been found which is as bad as it gets. We'll talk about that. No patch yet, but one is in the works. And of course the problem is lots of people won't patch.

We also have some welcome news from Yubico. We'll touch on a couple miscellaneous media tidbits just because our listeners know that I like some things that are on TV, and I'm an avid sci-fi reader, and John and I have just recently been exchanging email about something that's about to happen. And then we'll finish by taking a look at what's going on with biometrics, these recent high-profile failures and what it means. So I think another great podcast for our listeners.

JASON: And no surprise about that. And I should also mention that when you were talking about the upcoming sci-fi news that you'll be talking about later, John's fist was raised in the air. So he's very excited. He's very excited.

Steve: Since you're able to see John, is he aware of Ryk Brown's - okay, I heard him.

JASON: Oh, yeah. The thumbs-up.

Steve: Okay.

JASON: An emphatic, "Oh, yeah" and thumbs-up.

Steve: Okay, good.

JASON: I, however, oh, no. I'm sorry, I don't.

Steve: Well, you have children. So, yes. You're otherwise occupied.

JASON: I'm distracted a little bit. Although I did see the trailer for the next Star Wars earlier today.

JOHN SLANINA: The last.

JASON: Well, okay, sorry, the last "Star Wars," until the next trilogy that they decide to do.

Steve: And as a matter of fact, because I was curious about the IMDB rating for one of the things I'll be talking about today, I also went over to IMDB and watched the trailer for the final Star Wars. It looks fantastic, actually.

JASON: Of course. It's so dramatic. And I'm thinking as I'm watching, you know, my kids are six and nine; but, yeah, they've seen both of the previous two. And I'm just thinking, all right, apparently I'm going to do the thing where we watch them in order again, leading up to when we go to the theater, and see how that goes. So I'm looking forward to it. All right. Make it rain, Steve Gibson.

Steve: Well, our Picture of the Week was just - it was a fun one that I encountered when I was pulling the show together. It's not something that most iOS users ever see, which is the screen of an iPhone showing a bunch of OS booting sort of stuff. Anybody who's ever booted a Unix or Linux is used to seeing all of this amazing cruft scroll up the screen. That's not something you ever experience on an iOS device. So anyway, this is a screenshot of a jailbreak in progress showing the ELI stage 3 screen of text with a bunch of mumbo jumbo on it that doesn't really matter. And then it does have the logo overlaying the Apple icon showing clearly a king up and then a king that has been knocked over, because of course flipping the king over is the famous acquiescence to a checkmate in chess.

So anyway, it also shows that the phone is tethered, which is one of the requirements for this forthcoming jailbreak for iOS devices. And I've never had the occasion to jailbreak any of my i-things. Just I haven't had any need. I've been content with them the way they are. But I have an iPhone 6 still, and I was forced off of it when Apple played those games of slowing down the 6 in fear that the battery was getting old, and it was just so frustrating that I finally said, okay, fine, and I got a 10, which is a smaller phone than - I have the big 6+. And I'm happy with it, but I kept my 6 because then Apple figured out, oh, I guess we ought to check to see if the batteries are actually bad, rather than just assuming they are, because of course I had been babying mine for its entire life, and it was in perfect shape.

So anyway, the 6 came back to life after an update, and I still have it. So I will probably, since it's got one of the chips that's right in the sweet spot of this forthcoming, presumed forthcoming consumer jailbreak for iOS, I think I'll probably play a little bit with that phone since it's a spare.

JASON: You might as well.

Steve: Yeah.

JASON: You might as well, if it's just sitting around collecting dust. Now, so this is a tethered jailbreak, so does that mean that it will not survive a reboot? You reboot this thing, it's going to clear it out, so you're going to have to tether it again in order to do that again?

Steve: Correct. And that's fine for me because I would only do this just sort of as a lark in order to see what it was like and maybe screw around a little bit. But so it's a tethered break. It intercepts the boot process in a way then that will allow a full jailbreak to be developed, which Apple for that range of devices, from the iPhone 4S, I think it was, that used the A5 chip, all the way up through actually the iPhone 10 that I have, would also - it also uses the A11, which is the last one. Apple found the problem last summer and fixed it. So for the XS and all the other phones and everything since, the A12 and the A13 devices, this problem's not there. So the problem's been known for a year, but no one had figured out how to leverage that into a boot time jailbreak, and that's now been done.

So an update, because this was the topic of last week. There is a site where this jailbreak will probably first appear. What's odd is that the name is the hacker-ish name, not the name that you would expect. So the site is Checkra1n.com. That's the site where we're expecting to see this appear. And in fact I'm going to just check. I didn't look - oh, yeah. So I just brought it up, Checkra1n.com. It's just a placeholder page with nothing but that logo that I mentioned seeing on our Picture of the Week, which is the king upright and then I guess it's the black king behind it knocked over. The exact spelling is important because bad guys immediately jumped on the normal spelling of the site, Checkrain, R-A-I-N, and put some bad stuff on that site. So I wanted to caution our listeners, anyone who may be interested in going to the Checkra1n spelled R-A-1-N dot com, that is, that's the proper site. That's probably where this is going to appear.

And also note that it's still the case for our browsers that, if you just put the domain name in and hit Enter, the browser assumes non-HTTPS. Which to me that's interesting because of course the most recent metrics are showing that 80% of Firefox users - we'll be talking about this a little bit later - 80%, eight zero percent of Firefox users are now at HTTPS-protected sites by default, or by habit, 80% of sites are now HTTPS as opposed to not. At some point it would make sense to me that browsers would start assuming HTTPS if the user just puts the domain name in. But that still hasn't happened. I put it in. I put Checkra1n.com into Firefox last night, hit Enter. Up came this placeholder page. Then I was curious. So I manually put https:// in front of the domain, and I got an error, which was odd.

Well, it turns out that the site is, like, again, it's sort of not like official yet. So it's got a bogus Let's Encrypt cert configured on that domain to a site. The cert that comes up is Q-W-E-R-T-Y - of course "qwerty"; right? Q-W-E-R-T-Y. Then all lowercase, well, case doesn't matter for domain names, O-R-U-I-O-P. So then it's not quite across the top, that first row of alpha, but then it kind of ends up with the U-I-O-P. So let's see, I guess it's just they inserted an extra "O" in there, and then it resumes with Y-U-I-O-P dot com. Anyway, so I thought, what the heck. So then I went there. And what's there is that same qwertyoruiop, and then it says hyphen security researcher and a few other lines of text. But there's nothing else at that site either.

So I don't know what's going on. But that certificate is currently configured on this weird Checkra1n.com site. So your browser of course complains. I had to push through a bunch of "Are you sure" and, you know, blah blah blah warnings. So just put in, you know, just go http:// or don't do anything, and your browser will probably take you there. That's where it's going to appear. It hasn't yet appeared. But it looks like that's going to happen. There's been a lot of dialogue online. The hackers are scurrying to leverage the CheckM8 vulnerability which is still up as sort of a proof of concept up on GitHub. I imagine within a week or two, I mean, these things, it may take a while, but everybody believes we're going to soon have a jailbreak that Apple will not be able to fix. So that's the site to keep an eye on.

And certainly, if that happens by next week, you and I will be talking about it; or the week after, you and I will be talking about it; or the week after, you and I will be talking about it; or the week after, you and I will be talking about it. I have a feeling you and I are going to be talking about this.

JASON: Probably so. It's really unfortunate that this isn't the other way around because everybody's going to hear "Checkrain" and type it in the normal way and then, you know, be like, oh, I'm going to install whatever that is.

Steve: Yes, yes, yes. Isn't that...

JASON: It's really unfortunate.

Steve: Is that dumb that the hacker, like, did the hackeresque name, or didn't get both of them. I mean, you know, dotcoms are cheap now. And so it's like, wow, that just seems like so unfortunate because what is at Checkrain is, I mean, it's not malware, but it offers some kind of semi-malicious pay-to-load junkware apps which unwitting users are going to grab thinking that it's the jailbreak, and it's not. Anyway.

JASON: Indeed, yup.

Steve: As we know, steganography is the practice of hiding in plain sight. Well, kind of. A better way to put it would be adding unrelated - well, at least in the digital era. You could certainly have used steganography back in Picasso's day to hide some image in another image. But, you know, now that we're in the digital era, you would call it adding unrelated data to an existing file in such a fashion that the original appears to be unchanged while still carrying that unrelated data. And we've talked about image files having that done to them. JPEG and PNG image formats have had steganographic content added to them while still showing the original image. With JPEG, it's very tricky because JPEG itself is deliberately a highly lossy compressed format. As we know, it achieves very high compression of images by representing the original image in the frequency domain using a technique known as "discrete cosine transforms."

So the point is that what's put in there, the original image that's compressed is not at all what comes back out. But it visually looks similar to our eyes. By comparison, PNG images, when they are not using a reduced color palette, which they can be palettized images, which then approximates the exact colors in the image with a reduced color set in order to get additional compression. But when it's not using a color palette, when it's using full 24-bit color depth, those images are inherently lossless. And so that makes hiding data in their least significant image pixels much more straightforward.

Anyway, what security researchers have now detected in use in the wild is they're calling them "malicious WAV format audio files." And I guess it's malicious content. But the WAV format will not, you know, it's a non-executable format, very much like images. So it's not like playing the WAV format would, unless there was a flaw in the WAV codec, which the WAV format was leveraging, playing the WAV format would not itself execute malicious content on your system. The idea is that, for some reason, the bad guys are conveying malicious content hidden in a WAV file. And in some cases the WAV file still sounds fine. I mean, it seems unaltered. In other cases it just sounds like noise, like static.

So it turns out that they've found them both using least significant bits in the audio sample, in which case our human ear would not notice a few, maybe it's even more than one. We would not hear a little bit of Least Significant Bit noise in your typical audio playback. In other instances they're just putting a WAV file header on the front of a traditional binary format file. And when you play that back it just sounds like white noise. It's just nonsense. So anyway, Symantec researchers spotted the Russian-backed Turla threat group, also known as Waterbug or Venomous Bear. No one seems to be able to agree on the names of these Russian guys. They all make up their own names, and then they later realize, oh, those are the same people that somebody else was talking about.

Anyway, they're delivering the publicly available Metasploit Meterpreter backdoor embedded in a WAV file audio track. And then subsequently Cylance found that the same steganography method was employed to infect targeted devices with the XMRig Monero cryptominer or Metasploit code designed to establish a reverse shell. But again, you have to have some malware on the receiving end that knows to look inside the WAV file to obtain this content. So it must just be that the bad guys have chosen to hide stuff in these formats because they are not typically scrutinized by firewalls or intrusion detection systems. Those intermediate filters see a WAV file and go, yeah, okay, fine. They're not listening to it.

Now, in the case of this one that's completely nonsense, it would be possible to listen to it, kind of, like to see whether it looks like a WAV file because visually we're able to examine the samples in a WAV file and see complex sine wave frequencies that are obvious. And what a binary file looks like is just absolute nonsense, absolute gibberish, clearly non-audio content. So if anyone really cared, it would be possible to discriminate that kind of a WAV header on the front of a binary file as a means for getting it through some sort of an intrusion detection system. On the other hand, technically it's a valid WAV file. So you'd have to decide if you really wanted to block something. I mean, maybe somebody wants white noise WAV file for some purpose that's legitimate and isn't in fact nefarious in content.

So anyway, the researchers took these things apart, did determine in the audible one, the one that still remained playable, that there was Least Significant Bit steganography occurring, and that the receiving code was extracting the least significant bit, reconstructing a file that was of interest to it, and then doing something with it. In the other instances, they were employing essentially a one-time pad based on a pseudorandom number generator to scramble the file, which would then be descrambled by XORing it with the same pseudorandom pad output on the receiving end in order either to decode it into a standard Windows PE format executable or to execute shell code. So just another interesting instance of steganography that we've not seen employed before in a WAV file.

And I suppose, again, if intrusion detection systems were concerned about this, it would be unlikely that they could detect it as LSB, as Least Significant Bits. But certainly, if something doesn't in any way look like it would be audibly playable, that is, it's just binary noise, I could see that an IDS might just decide to block it at the perimeter because it would be a means of getting a high-density executable file past a perimeter firewall and safeguards.

JASON: My question on that is what exactly is the difference between binary noise, just jumbled binary noise, and real white noise? They probably sound the same.

Steve: Yes.

JASON: And there are reasons to have audio files that are just white noise.

Steve: Yes.

JASON: Yeah, I wonder if there's any sort of difference that could be detected along the way between the two.

Steve: Well, so I've spent a lot of time in my life looking at binary executable code. And generally you've got, I mean, there are definite regions that are not white. Normally one of the reasons people compress Windows PE format files is they notoriously have huge blocks of zeroes. Just, for example, data regions in executable files are initialized to zero. They're not left uninitialized. They're all set to zero. The advantage is, if you make the mistake of using that as a pointer, you'll get a null pointer. And that immediately, that's guaranteed to cause a crash rather than, like, well, it depends upon what data was already in RAM when the file loaded. That never happens.

So you definitely have blocks that are non-white, that is, that are all null. Of course that would be silence in a WAV file, which is also not illegal, so that could happen. So it would be - it's certainly the case that you could create a heuristic algorithm to examine a WAV file to see if it makes sense, and just to decide this is not looking like a WAV file. This looks executable.

On the other hand, we're also seeing them deliberately scrambling the binary content with a pseudorandom sequence, so that would turn it into complete white noise. By definition, the output of a random number generator is white noise. So it's absolutely uniformly distributed random samples. So if you XOR anything with that, you're going to get white noise output, which is always sort of oddly counterintuitive, but it's true. And then you re-XOR that white noise with the same pseudorandom data, and you recover your original non-random executable content, which is the way a lot of block ciphers, or stream ciphers especially, operate. So, interesting.

JASON: Yeah, very interesting. You've got like a threefer on Firefox coming up.

Steve: Yeah, we do. As happened with Chrome, well, and actually there's a nice reason. I considered putting the third story first just to increase the significance of the other two because the third story is about - and we'll get to it in a second - about how the German security arm, Germany's cybersecurity - I'm at a loss for words. Not foundation, authority - chose Firefox among four competing browsers: Safari, IE, Edge, and Chrome. No, not Safari. So it was IE, Edge, Chrome, and Firefox. Of those four, Firefox was the only one to qualify for all of the points that this German cybersecurity agency - "agency," that's the perfect word I was looking for - was looking for. But we'll get to that in a second.

First, we are currently at Firefox 69. And Firefox has been gaining in popularity. Leo has switched to it. I'm still with it. I've always been with it. Because it's turning out to be the one that is most - that's demonstrating the greatest concern for its users' privacy and security. Google is in the understandably delicate position of being all about advertising, and advertising is all about tracking. And so Chrome just, you know, which is a Google property, is kind of stuck there in the middle. Firefox isn't. So Firefox 70, seven zero, which will be released next - I think it's by the end of the year. I don't remember. I didn't have it in my notes here. But we're at 69 now, so it's the next major release of Firefox will be following the industry to change the way sites appear in the URL address bar.

Currently, under 69, we get a nice, friendly, green padlock for HTTPS sites. And for sites that are presenting an extended validation, an EV certificate, the company's full name is shown. All of that changes with the next major release, unfortunately probably forever. The padlock no longer is green. It's kind of a dark gray. And that's intended to signify that HTTPS is the new norm. No more green. Previously, only insecure pages with login forms would be shown an angry red slash through a black padlock. Now, with release 70, the next one, all pages, every page of non-HTTPS sites, which is to say just HTTP, will receive the angry red slash padlock, regardless of whether or not they are requesting sensitive information from their visitors.

In other words, HTTPS is, as I said, the new norm. And as with Chrome, the user will need to dig down into the URL window-shade dropdown if they're curious to inspect the type of certificate being presented by the site. That will be the only way in the future to see whether a site is using an EV certificate, that they've decided they're going to economize on the space in the URL.

Also, as we've discussed previously, it is the case, for example, that right now, under this current Firefox 69, if you go to my site, GRC.com, it says "Gibson Research Corp." in green. That's been the traditional EV indicator. It's been noted, however, that there's nothing to prevent some other entity from incorporating Gibson Research Corporation in a different state of the United States and validly registering an EV certificate that shows the same name, which would lead anyone to believe, because we've been around a long time, that they're Gibson Research Corporation with an EV certificate, when in fact they are, unfortunately, with the same name, because I can't prevent somebody else from registering a name outside of California. So that's what happens.

So I did inspect the dropdown of the security site I was visiting at the time. I was on an Internet security site, and separately I was shocked by the number of unblocked trackers and third-party tracking cookies that Firefox was showing it was permitting. So it turns out that I had not been keeping up with Firefox's changes of late. Firefox has been making some content blocking changes, strengthening many of the defaults that it uses. And in checking into what was happening, I realized that I should have been changing my settings along the way. Since I imagine that others listening to this podcast may be in the same situation, I wanted to give everyone a heads-up.

Previously, to get the strongest security settings, it was necessary to use Firefox's custom tracking blocking options. That's changed. Under Firefox's main menu, it's like the little three bars in the upper right-hand corner of the browser chrome, under the second line item is content blocking. I had earlier set it to "custom." And as I said, that was the necessary setting at the time to enable the stronger protections which were not on by default. But now they are on by default, and custom wasn't getting it done for me any longer.

So our users want to switch to "strict," which now and probably forever implements the strongest protection for users when they're visiting the 'Net by default. And because it can, and Firefox warns of this, it can cause some sites to have a problem and to break, then you can reduce your security and privacy selectively by site if you choose. But anyway, I immediately switched to "strict." Everything seemed fine. But I had not realized that in moving forward my previous setting of "custom" was no longer blocking everything as I would wish it to. And so "strict" will do that.

And as you said, Jason, we've got three things. The second is I'm 100% sure that those who have been pushing Firefox in the direction of increased security and privacy felt rewarded by the news that Germany's cybersecurity agency is now recommending Firefox as the most secure browser. And that recommendation is within Germany for all of the corporate entities and other Germany agencies that are following this advice. I cannot pronounce the name of this German Federal Office for Information Security. It goes by the initials BSI, which stands for Bundesamt fr Sicherheit in der Informationstechnik. I apologize to...

JASON: Yeah, I think that was pretty good. I think.

Steve: I apologize to the German speakers among us.

JASON: From no authority whatsoever I think you did a great job.

Steve: We will refer to them henceforth as BSI.

JASON: All right.

Steve: Which is, you know - anyway, yes. They tested Firefox, Chrome, IE, and Edge. Of those four, Firefox was the only browser to pass all of the minimum requirements for mandatory security features. Thus Firefox received their top marks during a recent audit which was performed by them. So this BSI tested Firefox 68. So that's the previous edition. We're on 69 now, as I mentioned. They tested Google's Chrome 76, IE11, which is also the current one, the current IE from Microsoft, and Edge 44. Importantly, the tests did not include other browsers, namely Safari, Brave, Opera, or Vivaldi, which I do think is unfortunate since several of them are even more security and privacy-centric than Firefox. But I have a sense, and our listeners will, after we look at what this BSI organization felt was important, that those browsers probably would have not made the grade either.

So the audit was carried out using rules detailed in a guide for modern secure browsers that the BSI updated last month, in September of 2019. The BSI uses this guide to advise government agencies and companies from the private sector which browsers are safe to use. The first edition was published two years ago, in 2017. Then it was reviewed and updated, most recently this summer. As a consequence, the BSI updated its guide to incorporate an awareness of the improved security measures over the last two years which have recently been added to our modern browsers.

So that includes things like, well, in fact HSTS is older than that, has been around for quite a while. That's of course HTTP Strict Transport Security. Also SRI, Sub-Resource Integrity. That's the provision for verifying that something you are requesting from a remote server has not been changed. CSP, Content Security Policy, that we've talked about extensively, which allows the web server to publish its policy which it wants the browser to enforce for the things that it subsequently requests. Also telemetry handling and improved certificate-handling mechanisms.

So according to the BSI's new guide, to be considered secure, a modern browser must satisfy a set of minimum requirements. And I'm going to tick off a 21-point list, but I think it's important and interesting to look at what this is. A lot of these things won't matter to end users. They sort of have an enterprise spin to them, which is again why I sort of wish that we had the other non-tested browsers placed into the same context.

But so here are the 21 things that the BSI felt was important: Must support TLS. Yeah, no kidding. I don't think there's a browser that doesn't. Must have a list of trusted certificates. Must support extended validation, EV certificates. Must verify loaded certificates against a certificate revocation list or an online certificate status protocol. So either a CRL or an OCSP. The browser must use icons or color highlights to show when communications to a remote server is encrypted or in plaintext. Connections to remote websites running on expired certificates must be allowed only after specific user approval.

Must support HSTS. Must support same-origin policy, and must support content security policy. Must support sub-resource integrity. Must support automatic updates, which is to say must support a separate update mechanism for crucial browser components and extensions. Browser updates must be signed and verifiable. Browser's password manager must store passwords in an encrypted form. Access to the browser's built-in password vault must be allowed only after the user has entered a master password. The user must be able to delete passwords from the browser's password manager.

Users must be able to block or delete cookie files. Users must be able to block or delete auto-complete histories. Users must be able to block or delete the browsing history. Organization admins must be able to configure or block browsers from sending telemetry and usage data. Browsers must support a mechanism to check for harmful content and URLs. Browsers should let organizations run locally stored URL blacklists. Must support a settings section where users can enable or disable plug-ins, extensions, or JavaScript. Browsers must be able to import centrally created configuration settings, which is ideal for wide-scale enterprise deployments. Must allow admins to disable browser-based profile synchronization features.

Must run, after its initialization, with minimal rights in the operating system. Must support sandboxing. All browser components must be isolated from each other and the operating system. Communication between the isolated components may only take place via defined interfaces. Direct access to resources of isolated components must not be possible. Web pages need to be isolated from each other, ideally in the form of standalone processes. Threat-level isolation is also allowed.

Browsers must be coded using programming languages that support stack and heap memory protections. Browser vendor must provide security updates no longer than 21 days after the public disclosure of a security flaw. If the primary browser vendor fails to provide a security update, organizations must move to a new browser. And, finally, browsers must use OS memory protections like Address Space Layout Randomization or Data Execution Prevention. Organization admins must be able to regulate or block the installation of unsanctioned add-ons and extensions. Whew.

So, frankly, I'm kind of surprised that any browser was able to qualify against all of those. And I'm proud and impressed that the browser I've chosen and am using was the one that did. So according to the BSI, Firefox, as I said, is the only browser to meet all of those requirements. There were at least eight areas where the various other three browsers failed. So to simplify things a little bit, I'll first note that IE failed all of these eight requirements. And I don't know if that's that important. It's still present, as we know, in Windows 10, though Edge is its formal successor and is today a far better browser.

So neither Chrome, Edge, or IE offer support for a master password. IE has no built-in update mechanism. Neither Chrome, Edge, nor IE offer an option to block telemetry collection. IE alone is lacking in its support for same-origin policy, content security policy, and sub-resource integrity support. And really, you know, in this day and age, if you can't support same-origin policy and content security policy, you're no longer in the game. So IE is probably still present only for legacy support in enterprise instances where they've got things that will not run under Edge and only run under IE. It really is time to move away from that. Fifth, neither IE nor Edge offer support for browser profiles and different configurations. And lastly, Chrome, Edge, and IE all lack a provision for what was called "organizational transparency." Whatever that is, Firefox has it.

So it was the one that Germany chose, the only one they are now recommending going forward. And again, since this is not a comprehensive list of all available browsers, I recognize that it's a bit skewed. But on the other hand, reading through that list, I would be surprised if any of the other ones were also able to qualify. It's sort of amazing that Firefox could because, wow, they really did set a pretty high bar.

JASON: All right. So, great. We're still gushing about Firefox. What else is awesome about it?

Steve: Yes, I get that.

JASON: Meanwhile I'm over here, like I'm a Chrome user. I'm like, okay, well, maybe I should switch over to Firefox.

Steve: Well, Chrome is the majority browser.

JASON: It is.

Steve: More people are Chrome users than anything else. And maybe Google will look at the things where it fell a little bit short and decide whether or not they care. Let's see. Master password. I don't know why it doesn't have that. Block telemetry collection, that would seem like a good thing, to offer it as a switch. Leave it, you know, leave it off by default, but give it to the user if they want to. And then organizational transparency, whatever the heck that is. So provide some, and then Google could be recommended in Germany.

JASON: Right.

Steve: So Mozilla has wisely recognized that its own internal HTML JavaScript-based browser pages could be subject to exploitation. We're often talking about about:config, which is just this unbelievable list of line items of tweakable things that is inside of Firefox. And in fact it's so big that you have to use the search bar in order to whittle it down to something manageable. So anyway, we're always talking about that. Mozilla's taken the wise precaution of preemptively blocking both inline and eval JavaScript on those pages to prevent any possibility of successful script injection attacks. They recognize that mistakes can happen. And they had to go through some hoops to rewrite those pages to be safe after, well, to still be functional after this block was put in place.

It turns out, and I wasn't aware there were this many of them, they have 45 internal locally hosted about: pages. About:config is the one we're always talking about. There's about:downloads. About:memory, which shows Firefox's memory usage. About:newtab is the default page that you get with a new tab. About:plugins - believe me, I'm not listing all 45, just the top few. About:plugins is a list of all your plugins, as well as other useful information. About:privatebrowsing opens a new private window. That's a handy shortcut. And about:networking displays various networking information. And there's another 36 on top of that.

So since all of those pages are written in HTML and JavaScript and render within the essentially unlimited security context of the browser itself, they are prone to code injection attacks that, in the case of a vulnerability, could allow remote attackers to inject and execute arbitrary code on behalf of the user, i.e., in things like cross-site scripting attacks. So to add a robust first line of defense against code injection attacks, even when there is a vulnerability, I mean, even in the face of a vulnerability, Mozilla took the preemptive act of blocking the execution of all inline scripts, which would then foreclose any injected scripts as part of that.

They implemented a strict content security policy to ensure that JavaScript code only executes when loaded from a packaged resource using the internal protocol. That required them to, as I mentioned, to rewrite all inline event handlers and moved all inline JavaScript that had been present in those pages out of line into separate packaged files for all 45 of those about: pages.

In their blog posting about this last week, they wrote: Not allowing any inline script in any of the about: pages limits the attack surface of arbitrary code execution and hence provides a strong first line of defense against code injection attacks. So when attackers cannot inject script directly, they will fall back. They'll tend to use the JavaScript eval function and similar regarded-as-dangerous functions to trick the target applications into converting text into an executable JavaScript script in order to achieve code injection. So in addition to the inline scripts, Mozilla also removed and blocked all of the eval and similar functions which Mozilla feels is another dangerous tool. They're not happy that it's in JavaScript, but they need to support it.

They wrote: "If you run eval with a string that could be affected by a malicious party, you may end up running malicious code on the user's machine with the permissions of your webpage and extension." And even Google feels strongly about this. Google said: "Eval is a dangerous function inside an extension because the code it executes has access to everything in the extension's high-permission environment."

So Mozilla rewrote all use of eval-like functions from system privileged contexts. They removed them all, and from the parent process in its codebase in Firefox. On top of this, Mozilla also added eval assertions that will disallow the use of eval functions and any of its relatives in system-privileged script contexts, and will inform the Mozilla Security Team of any unknown instances which the browser may encounter. So what all of this means is that, for we Firefox users, Mozilla is being very proactive about the security on behalf of their users. And of course in this day and age, as we know, it is very important for them to be proactive.

We talked about Microsoft's ElectionGuard open source election security system shortly after it was originally announced in May. We didn't know much then. There was a placeholder on GitHub with some aspirational text, not much more. The good news is we're beginning to see some movement away from, in general, the whole environment of Diebold-style closed, failed, and proprietary election systems for-profit model. That's the environment we've been in so far, where the election machines are purchased by states throughout the United States and even elsewhere. India, with its huge democracy, has the same problem. We're beginning to move towards open.

So in my opinion Microsoft is helping this hugely. Last May 6th, during their Build Developer Conference, they announced for the first time their free open source SDK called ElectionGuard, which is aiming to enable end-to-end verification of voting. All of the goods are now up on GitHub. I have a link to them in the show notes. I imagine if you just google "Microsoft ElectionGuard" - as one word, ElectionGuard - "SDK," you will be taken there. It can be integrated into voting systems and has been designed to enable, in their words, end-to-end verification of elections, open results to third-party organizations for secure validation, and allow individual voters to confirm that their votes were correctly counted. So we're talking, I mean, this is a next-generation in - it uses state-of-the-art homomorphic encryption in order to pull off some of this magic. But that's where we need to go.

So it's back in the news because Microsoft has followed through by wrapping their bug bounty program in the major ElectionGuard modules. Not all of them yet because I guess they're not feeling that they're all quite ready to be attacked by the hackers of the world. But they are inviting security researchers from across the world to help with the discovery of high-impact vulnerabilities in this now-published ElectionGuard SDK.

In their blog posting announcing the bounty, they wrote: "The ElectionGuard Bounty program invites security researchers to partner with Microsoft to secure ElectionGuard users, and is a part of Microsoft's broader commitment to preserving and protecting electoral processes under the Defending Democracy Program. Researchers from across the globe, whether full-time cybersecurity professionals, part-time hobbyists, or students, are invited to discover high-impact vulnerabilities in targeted areas of the ElectionGuard SDK and share them with Microsoft under their Coordinated Vulnerability Disclosure program."

So ElectionGuard Bounty offers cybersecurity researchers a reward of up to $15,000 for eligible submissions with a clear and concise proof of concept to demonstrate how the discovered vulnerability could be exploited to achieve an in-scope security impact. In other words, for these things that are covered, Microsoft believes they are ready. The ElectionGuard components that are currently in scope for bug bounty awards include the ElectionGuard API SDK, the ElectionGuard specification and documentation, and the verifier reference implementation. Microsoft indicated that it will update the ElectionGuard Bounty scope with additional components to award further research in the future.

So this is just very cool. I can't think of anything more important than a respected entity that has been working with other cybersecurity organizations to develop a next-generation capability. It's free. It's open source. Why wouldn't proprietary vendors incorporate this with the understanding that they need to open their systems? There's nothing wrong with selling for profit an appliance that delivers the required technology. But the technology itself has to be open to audit and verification. That's just the way we have to go. So this is a great step in that direction. And I really - I don't often congratulate and salute Microsoft for things they do. Everyone on this podcast knows that. But in this case I think this is just wonderful.

JASON: Yeah, I completely agree. Definitely an important thing for Microsoft to be helping out with. But those behind these closed systems, like do they realize that it's important to open the systems? And I guess that's the ultimate challenge, right, is to get them to [crosstalk].

Steve: I think that remains to be seen. Yeah. I think what's going to happen is that, while all systems were closed, the purchasers in states have had no choice but to choose among the best what they felt was a closed solution. Now that we have open solutions - and there have been some rumblings that states would no longer purchase any closed solution. So that's what we have to have. As soon as one of the major vendors switches to ElectionGuard and offers a commercial implementation of Microsoft's open and auditable ElectionGuard system, then it'll be possible for the purchasing entities spread across the U.S. to say, okay, we're not buying it unless it is an open auditable ElectionGuard solution.

And the other cool thing is it's not just that it is open and auditable. It offers features that voters want. The idea that you are able to, after the fact, go home and take a receipt that you received from your voting experience and put it into a website and interactively, with cryptographic verification, know that your vote is among those counted and tallied, that's super cool.

JASON: Absolutely.

Steve: I mean, you know, that's just neat. That does away with all of the backroom shenanigans, all of the problems that we've had in Florida with hanging chad and all of this, I mean, it's a new game. So it always seems that, in order to fix things, this country has to go through a period of intense pain and anguish. And, you know, then we get things fixed. So we've had that with our voting systems in the past, and it really does look like the consequence of that is a whole new dawn of voting integrity. So yay.

JASON: Right. Yay, agreed.

Steve: Okay. So of crucial importance to anyone, well, of crucial interest, I won't say "importance" because I don't want to overstate the actual impact of this, but the impact will be up to individuals to determine for themselves, given the set of facts. And the set of facts are without controversy. As we know, kernel driver flaws are always worrisome because they can be extremely potent, being in the kernel. And flaws are even more problematic when they affect wireless interfaces. And they are more problematic when they affect systems that are in their default configuration. And they are still more problematic when they require no user interaction to be exploited. And they are even more problematic when the flaw is a buffer overrun of a remote attacker-provided buffer of data.

This bug being tracked as CVE-2019-17 - and I love the fact that it ends in 666 - has all of those properties. It was discovered by Nico Waisman, the principal security engineer at GitHub, last week while he was examining the handling of Notice of Absence protocol packets in Linux. A patch to correct this is currently under revision, but has not yet been incorporated into the Linux kernel. And I checked just before the podcast. So hopefully, by the time people listen to this, that may change. But so it's like it's very pregnant right now. If successfully weaponized, and to our knowledge, to public knowledge that has not yet been done, if successfully weaponized, it would allow attackers to fully compromise vulnerable machines remotely in their default configuration without any action on behalf of the machine's user.

It is classified as "CRITICAL," in all caps, in severity, for all of those reasons. It exists in the "rtlwifi" driver. So it only affects Realtek WiFi-based systems. The driver's been found to be vulnerable to a buffer overflow attack. Obviously, systems lacking WiFi or with their WiFi disabled or with some other non-Realtek WiFi chip, will all be safe. Otherwise, not so much. The driver flaw can be triggered when an affected device is simply within radio range of a malicious attacker's device. As long as the WiFi is turned on, it requires no interaction on the part of the end user.

The malicious device would exploit the vulnerability by using a power-saving feature known as Notice of Absence. It's built into the Wi-Fi Direct protocol. Wi-Fi Direct is a peer-to-peer standard that allows two devices to connect over WiFi without the need for a common access point. The attack would work by adding vendor-specific information elements to WiFi beacons that, when received by a vulnerable device, would trigger the buffer overflow in the Linux kernel. So once again, it only affects Linux devices using a Realtek chip when WiFi is enabled. It cannot be triggered if it's turned off or if it uses a non-Realtek chip because that would have a different driver without this flaw.

Reporting does indicate that Android devices containing Realtek WiFi chips will also be affected. So of course that's big because Android devices are inherently mobile. They inherently have WiFi turned on. And many of their kernels are unfortunately never going to be updated. This has been...

JASON: I've been there, yup.

Steve: Yeah. This has been - I've seen some reports saying 2013, which would make it six years old, although more credible reports are saying four years old. So at least for the last four years. On the other hand, that is a ton of Android mobile devices that have been sold by vendors that are not updating their kernels ever. So this is potentially a huge golden nugget opportunity for the bad guys, if they can figure out how to weaponize this.

So far, all we've seen disclosed publicly is a denial of service, which crashes the targeted device. So we don't yet know for sure that anything more is possible.

However, looking at the source code - and it's easy to do. Everybody, all the bad guys are looking at it right now. It very much looks like a user-provided packet is the thing that will be made to overflow the buffer. And that, I mean, that's the golden keys. That's what you want. The flaw was introduced in the driver innocuously.

The Linux gurus are all over the case. Laura Abbott posted to the Linux Kernel Mailing List a note saying: "Nicolas Waisman noticed that, even though" - and the variable is N-O-A, that's Notice of Absence - "noa_len," that is, the length - "is checked for a compatible length, it's still possible to overrun the buffers of p2pinfo, since there's no check on the upper bound of noa_num." In other words, the noa_len is probably the length that is self-declared in the packet. I didn't look, but that would make sense. Whereas noa_num is the actual size of the packet, and there's no check on it.

So then she says: "Bounds check noa_num against P2P_MAX_NOA_NUM," which essentially is like adding one simple test to the code. In other words, it is trivial to fix this. It's not going to take any time at all. They're already, I mean, the fix is known and has been for a week. The problem is it hasn't yet been pushed out; and, even when it is, how many systems with Realtek chips enabled of any sort are not going to get that patch?

So anyway, security-conscious Linux users with Realtek WiFi will want to be on the lookout for a Realtek driver update which should be coming very soon. This is not a mysterious or difficult-to-fix patch. I'm sure we will have it shortly. If you are really security conscious, you might want to disable WiFi, if you don't know that you need it on. There is no way to protect yourself currently with an unpatched Linux kernel and Realtek chips if you need WiFi on. Even if you've established a link to a local access point, your chip will still promiscuously accept an incoming packet on this WiFi direct peer-to-peer protocol that would allow it to be taken over.

Again, this has just happened. So there is no known - there's no publicly known takeover. But I have a feeling, Jason, you and I in the next week or the week after or the week after may be talking about that having been developed. And it'll be interesting to see if we get more news on the Android front because of course that's the big enchilada here because there are, you know, a lot of laptops have WiFi on. Hopefully they will be able to get updated. Linux-based servers probably don't have WiFi enabled. They're all going to be safe. But the big target are the Android devices, if they are known. Do you have any sense for the presence of Realtek chips in WiFi in Android devices?

JASON: Well, you know, as I was reading this earlier today, I was doing some Google searches to try and find some sort of list or some sort of indication what devices have this in there. And I couldn't find any exact list that detailed devices, just that they exist in many. And so there's the challenge is it's kind of obscure to even know that it's there. Possibly there's a way to go into system settings and be able to look for a certain piece of information or detail in there that would show it on a device-by-device basis. But I couldn't find any indication.

That's what really worries me is that if it's so obscure to know this, and that coupled with just the bad track record of updating the Android OS, at least on a Linux-based system you can apply an updated driver, and boom, you're done. On Android it's really just like the manufacturer or the maker has to roll out a separate security update to the OS entirely. And they're just not motivated to do that. They're really not.

Steve: No.

JASON: Especially on four-year-old devices. So as with everything in Android security, when things like this happen, I've just over the years just, I think, become a little jaded and just realized, like, you know, here's yet another thing that's not going to be patched. Google is going to maybe do some sort of update to address it. But who's going to get that? Not many.

Steve: Well, yeah. In fact, our advice for quite some while after this pattern of lack of third-tier manufacturer updates was clear, was if you want an Android phone, get one from a mainstream supplier, that is, like Google, Samsung, you know, one of the big guys that is taking the maintenance, the aftermarket maintenance seriously. Because, you know, boy, it's necessary.

JASON: Yeah.

Steve: And this sort of problem isn't found often. But, boy, I mean, this ticks off every single checkbox of the worst possible nightmare. It's no action on the user default configuration; no connectivity needed; kernel exploit, the driver is in the kernel so you have full access to everything in the machine. It doesn't get any worse.

JASON: Yeah, and I think you're right, like Pixel, Samsung phones. Say what you will about Google's Pixel lineup, they're not as popular as so many other phones, but they're almost guaranteed to get the updates as soon as they're available. And to that end, you know, they end up being the safer pick. But do Pixels have this in them? Do they have the Realtek WiFi? That's a great question. I guess we'll find out. This is one of those things that we'll keep an eye on over the next couple weeks.

Steve: Yup. Well, and we're glad to have you on the podcast for the next few weeks because you will be the guy to know.

JASON: We'll see, yes. I'll help keep you posted.

Steve: So Yubico, the company our listeners know well because I've spoken of them from, like, before anyone knew of them, actually, they've been working toward a local Windows logon solution for quite a while. It was originally made available in a non-official preview form, as long ago as last March. It's been available for Mac and Linux machines. And I am very pleased to announce that Yubico's solution for local hardware dongle-protected login to Windows machines, Win7 through 10, is now available. In fact, I recently took a look at it. It was a few months ago. It was before I took my laptop out of the country for the SQRL World Tour. It is very clean and simple. It is now available.

So remember that, to be useful, the hard drive also needs to be encrypted with BitLocker or whatever your preferred encryption tool is, so that the drive cannot be removed from the system and mounted on another system and then just simply read. So you want to encrypt it. As long as you do, if you really want strong local security, that is, no active directory support, no enterprise stuff, it's just your Windows-based laptop, your Windows-based desktop, you can now use an inexpensive and very secure YubiKey to make that possible. And it's a free download. I have the link in the show notes, on page 10 here of the show notes, if anyone is interested. But it's also easy to find it just by cruising around Yubico's website and grabbing it.

JASON: Right on.

Steve: Two quick short random miscellany bits. I am a huge longstanding fan of the Jason Bourne movies. So I was very curious and hopeful to see what the USA Network had in store for us with last Tuesday's release of the first episode of their new "Treadstone" series, which is set in a time that seems well after the Jason Bourne adventures. At some point they refer to him in the past tense. This first season will have 10 episodes, each releasing on a Tuesday.

So this evening, since this is Tuesday, I will be checking out, well, I may actually rewatch the first one again, which I did see last week. But it jumped around a lot, and it treated me as though I was smarter than I apparently am. In other words, it left me feeling a bit confused. But I'm going to watch it again, but I'm probably going to watch the second one. So the jury's still out, but I wanted to bring it to the attention of any other Jason Bourne fans, since it does look like it may develop into something useful.

And speaking of useful, the long-awaited and very much anticipated second book in Peter Hamilton's newest Salvation trilogy releases one week from today, on October 29th. As I mentioned, I was exchanging emails with John Slanina in the last couple days because he mentioned he had reread the first one. I need to reread it, too, because it's been a while. That's the only problem with Peter Hamilton stuff is that the books come out few and far between. They are incredibly good, I mean, he is my absolute number one favorite science fiction author, period. But there just isn't enough of him.

The timing is perfect since I just wrapped up a very enjoyable five-book series. John, if you're listening, it was the Terran Fleet Command Saga by Tori Harris. Five books. They're not Hamilton grade. But then in my opinion, as I said, nothing else is. I would recommend the Terran Fleet Command Saga to anyone who enjoys a nice, well-written, slow burn, well-assembled, space opera combat mystery involving some interestingly enigmatic aliens. The series had many great moments.

And of course all of this is set against the background of Ryk Brown's ambitious and ongoing 75-book Frontiers Saga series that I know John is also reading. After finishing the Terran Fleet Command Saga, I quickly read book 12 of the second set of 15 which had recently dropped while I was reading the other series, to see how all of the people that we've come to know so well in Ryk Brown's series are faring. So I'm of course following that.

You know, each of these authors has a very different flavor and style. And if asked, I guess I would be unable to choose among them. I've never encountered any other author with Peter Hamilton's storytelling and reality creation talent. But for the sci-fi content-starved, Ryk Brown's Frontiers Saga, which just is producing wonderful pulp sci-fi, is just so much fun. So anyway, just for those who are sci-fi enjoyers, and I know many of our listeners count themselves among that set, I wanted to make a note that the second book in the Salvation trilogy is a week from now. I will start rereading the first one just because it's such an enjoyable read. And then I'll know that number two is ready. And then everybody is going to be frustrated while we wait for the conclusion, you know, what, another year probably. But they're worth the wait.

And speaking of worth the wait, Jason, let's take our third break and then talk about the biometric mess that we find ourselves in.

JASON: This episode brought to you by Steve Gibson. We don't have a third sponsor today, Steve. It's all good.

Steve: Oh. That happened last time.

JASON: Actually, how about this? Twit.community. If you didn't know we have a forum, go to twit.community, and you can talk with other TWiT fans who are on the forum, just basically talking about topics from the shows, talking about pitching in with the hosts of the shows, about the shows themselves. So twit.community. That's the ad. That's the third.

Steve: You know, it's funny because Leo is so excited about this.

JASON: It's really cool.

Steve: Well, GRC has had newsgroups forever. I mean, the GRC newsgroups predate this podcast. That's how old they are. And they are. They're NNTP text-only newsgroups. But, I mean, so I'm well aware of the power of a true useful community. And I've heard Leo remarking at how amazed he is at the quality of the people who are there.

JASON: Oh, yeah.

Steve: And although the groups do require moderation because you want to keep bots and ads and despoilers out of them, it is so useful to have them. All of the SQRL project was done in sight, and also with significant help from many other people in the newsgroup. And really they're in the process of taking it out of my hands at this point, as they should, because I will soon be returning to SpinRite, and all of that development was being done and will be done, again, in plain sight in GRC's newsgroups.

And of course I recognize that there is a place for web forums, which is why SQRL also has its SQRL web forums at sqrl.grc.com, where sort of a different demographic of people prefer to hang out. So it's definitely the case that there's a place for this. So I'm really, really delighted that TWiT got some. That's really very cool.

JASON: Yeah, it's been a lot of fun kind of getting in there, and every once in a while I'll get an email, an autofire email from the community saying, hey, someone mentioned you. It's just nice to have that open conversation, jump in there; and, you know, it's a good time.

Steve: Okay. So the mess of biometrics. We had two failures in biometric authentication which hit the news this week. So as I mentioned at the top of the show, I thought this would be an opportune time for us to check in with biometrics to see how it's all going. Sophos had a story about - well, many people did, but I liked their coverage - about the newly, well, actually it's not even newly released, is it. It's like the Pixel 4 is supposed to start shipping Thursday.

JASON: Yeah, it's not even out yet.

Steve: In two days. So it's not even - exactly. They opened with the rhetorical question, does it matter that Google's Pixel 4 Face Unlock works even if the owner has their eyes closed? To which most of us, after thinking about it for a second, would answer, uh, yes.

JASON: Yeah.

Steve: That does matter. Turns out that the Pixel 4 is following in the footsteps of Apple's Face ID technology, similarly dropping fingerprint recognition in favor of optical face recognition. But Chris Fox, a reporter for the BBC, actually he wasn't the first to - no, no, he was the first to bring this to light. I was about to confuse this with fingerprint problems we'll get to in a second. He discovered a potential issue, which is Face Unlock works when the user has his or her eyes closed, like when they're asleep.

It turns out it's not necessary to get Google to confirm this, since it's already on the Pixel 4's help pages. Google wrote: "Your phone can also be unlocked by someone else if it's held up to your face, even if your eyes are closed. Keep your phone in a safe place, like your front pocket or handbag." Now, what's interesting is that there were some leaked images of the Pixel 4, and I've got one here in the show notes. It's been noted by the press that the leaked images show an option under Face Unlock which reads, and I'm reading it off of the show notes: "Require eyes to be open." And then underneath it says: "To unlock the phone, your eyes must be open," and there's a switch. Which, now, what's interesting is that in the leaked screenshot, the switch is off, which seems odd because this seems like an obvious good thing...

JASON: Totally.

Steve: ...for Face Unlock to have.

JASON: I don't know why you'd have this feature and don't just default to that always on. Like don't even give the option on that one. It's not necessary.

Steve: I agree. It's like, okay, now, what is the use case for wanting to unlock your phone when your eyes are closed? It's a little difficult.

JASON: Maybe it's faster because it doesn't have to detect that your eyes are closed? It's just like, oh, we've got all the - I don't know. That's the only thing I can think of. Like if you want it just slightly faster. But who knows if that's true.

Steve: Yeah.

JASON: Who knows if that's true.

Steve: And so the obvious risk is that anybody could get hold of your phone - your kids, your spouse or your partner - and without your knowledge unlock it while you're sleeping or passed out or for whatever reason you're unconscious, and when your eyes are closed, and have access to your phone. So what's interesting is that this appears to be unintended behavior because what's happened is the devices that were actually released to the press, both the BBC and the Verge, have preview devices, and that option has been removed. The switch is no longer there. And it will presumably not be there in two days. In fact, Google has stated that it plans to fix the issue "within months." But they have not been more specific.

In the meantime, anyone who is concerned about this, Google recommends using a PIN or an unlock pattern. In other words, if you're worried that your phone, your brand new Google Pixel 4 with Face Unlock, will not pay attention to your eyes - and apparently the option is gone from the option screen, and I'm guessing it's because it didn't work very well. That's the only reason I can imagine it was there. Maybe they hoped to get it working by the time it was released, so it was there initially disabled, maybe, because it wasn't available yet, but they hoped it would be. And then it was like, okay, we've got to ship this sucker, and it's still not robustly working, so let's take it off the option screen and just tell people not to use it if they're concerned. So basically at ship, in two days, the phone will, we believe, not have this option. Google confirms that now. And if that's a problem, don't use it.

JASON: Yeah. And, I mean, this is not something that Google hasn't done before. Like actually very recently, with the release of Android 10, when that software launched, people noticed right away that third-party launchers could not use the gesture navigation. And Google had to admit, like, yes, you know, we really wanted gesture navigation to work with third-party launchers. Lots of people use third-party launchers. But we just ran out of time and had to go with this. So at a future date that compatibility will be added. And actually in the Pixel 4 that compatibility is added. But this is something that Google will do. They'll run out of time and - maybe lots of companies do this in different ways. But Google definitely does.

Steve: Ship it anyway, yup.

JASON: And in the case of security and something like this, you just, like, what? I don't understand how you have this feature, and you don't have an eyes-open requirement. At the same time, I mean, and I'm curious to hear your take on this, when it comes to biometrics, and like say I fall asleep, and my phone's next to me, someone could very easily take my phone and put it on my finger for the fingerprint capture, as well; right?

Steve: That's true. That's true.

JASON: So that's kind of the same thing.

Steve: That's true, yeah. Well, and speaking of fingerprints, it turns out that the - thank you for the segue, Jason. It turns out that the sexy ultrasonic finger reader used by Samsung's flagship S10 and Note 10 smartphones can be spoofed with a $3 screen protector.

JASON: Wah wah wah.

Steve: At least one that's not made by Samsung. This recently came to light, although it was not the first time it was observed. But it made news when a British woman claimed, and others including Samsung have since confirmed, that after fitting her new phone with a screen protector, she was then able to unlock her S10 using any of her fingerprints, including ones not enrolled in the phone's authentication system. And since that made her curious, she reportedly then asked her husband to try the same thing, and his thumbprints worked, too. As did the same trick on her sister's Samsung phone. So obviously something was wrong. Samsung's initial response when confronted with this was "We're investigating this internally. We recommend all customers use Samsung-authorized accessories, specifically designed for Samsung products."

Then last week, in comments to Reuters, Samsung admitted the problem was real and said it would release a software patch. Now, that's going to be interesting because this seems like a technology problem. Anyway, they said: "We are investigating this issue and will be deploying a software patch soon. We encourage any customers with questions or who need support downloading the latest software to contact us directly."

So, okay. The issue with the S10 and screen protectors was first noticed when the smartphone was launched back in February of this year, but the issue failed to acquire critical mass. Unlike earlier designs, which used a dedicated sensor, the Qualcomm ultrasonic technology used by Samsung is embedded behind the screen and uses high-frequency sound, ultrasonics, which are modulated by the pressure of a user's finger against the screen glass. It was noticed, however, that covering the screen with a screen protector could in some instances create an air gap that could interfere with the integrity of the system's ultrasonics. Samsung's advice, which still seems a bit flaky to me, is to use its brand screen protectors that use, they say, special adhesives that eliminate the possibility of any gap. So to my mind, this whole technology seems sketchy at best.

JASON: Yeah.

Steve: I would say that ultrasonic trans-glass thumbprint reading is not compatible with screen protectors. That is to say, I mean, I think the technology is marginal. And like when you put your fingerprint, right, you press your finger directly on the glass. The idea that you can read an image through the display technology and all of its intervening electronics with some ultrasonic sensor behind there and get an image, to me that's already amazing. Then the idea that you're going to introduce an additional uncontrollable layer of goo, which is to say adhesive, and then a plastic laminate in between the glass that is being read, to me, you know, I wouldn't be at all surprised if this update actually checks to see if they are able to get an image and refuses to operate if there is no image obtainable.

That is to say, if in fact there's an air gap, then you're not going to actually get anything that looks like a fingerprint. And so probably the technology that existed just took whatever it was getting and said, okay, you've got a weird thumb, lady, but fine, and then just used it, when in fact it wasn't a thumb, it was an air bubble that it was resolving. And so anybody else pressing on it got the same air bubble. So maybe what they're going to do is be a lot more discriminating about does this actually look like a finger or not, and just say no. And then say buy a real screen protector, or sorry, we're - maybe they'll just decide our technology is not compatible with screen protection. Good luck. Get a good case. I don't know.

JASON: Yeah, we'll see how that goes.

Steve: Yeah. To me it feels like they've hit a bump, a big bump in their fundamental biometric technology. And really this sort of speaks to the larger issue that I just sort of wanted to address, which is that we're always talking on this podcast about the tradeoff between convenience and security. A long, impossible to memorize password is incredibly inconvenient, and incredibly secure, if you otherwise manage it correctly. If it is really long, and full of gibberish, you know, if it's high entropy, uses all the available characters, that's a really, really - that's really strong protection. And it can, although it isn't used to identify someone because we can't prevent them from using monkey123, if it actually were, like if it had 256 bits of entropy, it could be used to securely authenticate someone. I mean, it's that good.

Whereas notice that the biometrics we're using today, they're not identifying someone. They're confirming a presumed identification. That's a far lower bar to meet. For example, time-based tokens, the six digits that change every 30 seconds? Similarly, they don't identify you. They only confirm your presumed identity, which again is a much lower bar to meet. So the biometric systems that we have today, they're making a go/no-go decision. They assume who it is. They've been trained on someone's face. And so they're looking out at the person going, hmm, does this look like Granny or not, and making a go/no-go decision.

So what we're asking for is far lower determination, determinism, than having them look out and go, oh, that's John from down the street, and then loading his account or something. No. This is a simple yes/no decision being made. And even so, it is very problematic. Do you know, Jason - and I didn't track the technology in the Pixel 4. We know, because I was following it when Apple came out with their Face ID, all this fancy 3D camera stuff, how they paint a dot grid, an IR dot grid out on the person's face in order to require it to be 3D, and then they do a full multiple stereo vision rendering thing. Does the Pixel 4 have all of that?

JASON: Yeah. The Pixel 4 is the first Pixel to do that, actually. And I think that was part of, I mean, from what I understand, that was part of the reason why Google got rid of the fingerprint sensor altogether, because they felt confident enough in that collection. So it's very similar to Apple's approach. It sounds like one of the main things that's missing from the Pixel 4 in the Apple approach is the eyes-open detection, which is a pretty [crosstalk]. So, yeah, it's similar.

Steve: Yeah. And, you know, you would not think, given all the awesome technology that Google has in that phone, I mean, I watched their presentation. It was like, holy crap. I mean there's a lot of stuff in there. You'd think it could tell whether someone's eyes are open or not. I mean, a dog can.

JASON: Well, yeah. And what's interesting also, but maybe this doesn't have anything to do with the eyes, is there's another setting in Android settings called "screen aware" or something like that. And it's a setting that, when it's activated, it can tell when you're looking at your phone in order to keep the display from timing out. But now that I say that, like it might just be that you're facing the phone, not that your eyes are open. If your eyes are open, then it's like, okay, you've kind of already got the detection going on.

Steve: And that sounds like that was a pre-Pixel 4 thing, too; right?

JASON: Yeah.

Steve: The phones have had that for a while.

JASON: Some phones have had that. I don't know that the Pixel has had that, though.

Steve: Oh, okay.

JASON: Yeah, I think that's a new feature of the Pixel.

Steve: Interesting, yeah.

JASON: Yeah, I wouldn't be surprised if, in light of this, with the ultrasonic fingerprint sensor allowing anyone's fingerprint to let someone in based on the screen protector that's installed, if Samsung then ends up going, and maybe they're already working on this, going in the same route as Apple and the same route as Google with the Pixel 4, and this just becomes another reason why they then get rid of the fingerprint sensor and go for the face.

Steve: Do I remember Leo saying that he was not happy with the performance of the S10's sensor? Does it take a while to operate?

JASON: Yes. Yeah, yeah. I think Leo and I share the same opinion. And it wasn't just the S10 in display. Well, no, sorry, it was the Note 10 and the S10 both have the in-display fingerprint sensor. And it's the ultrasonic, of course, different from what we've seen a lot of other in-display fingerprint sensors be, which was optical. And like the OnePlus 7T and 7 Pro, they have an optical in-display fingerprint sensor that is super responsive and super quick and works almost every time. The S10 and the Note 10, in my experience, and I think Leo would agree, is just really, like it works half the time. It's hard to get it just right. You do it, and it's like, press a little harder, press a little - it doesn't work nearly as effectively.

Steve: Interesting.

JASON: Even though it's supposedly the better, more secure solution.

Steve: Yeah. So you are able to do an in-screen optical sensor. That's cool. I didn't know, I didn't realize that there were those around. I agree. That would be great. But again, everybody's got a camera already looking out the front of the phone. The problem, of course, is that it's 2D, and you absolutely need to have 3D in order to up the ante and try to determine whether this is a live or static person.

JASON: Right, right.

Steve: So everybody's being - and who knows. I mean, we've already seen 3D modeling of faces and all kinds of other stuff that you're able to do once you've got 3D camera technology looking out at you. So it just sort of seems like that's the direction everyone's going to go in. And of course the other problem that we didn't talk about, but we have in the past when we're discussing biometrics, is a super, super complex long password, which can be used to uniquely identify you and is incredibly robust and secure, can be changed. You cannot change your face. You cannot change your fingerprints. So on the one hand, your face and your fingerprint are something you always have with you. The problem is they're something you always have with you. And, like, for life, rather than just for your current excursion from the house.

JASON: I always tend to fall back on multiple points of authentication.

Steve: Yes.

JASON: Like I've always wondered why, if the fingerprint sensor on a phone is so ubiquitous - and now we're kind of heading out of that trend, it seems like, getting rid of it. But if it's so ubiquitous, why can't I have it so that I'm using my fingerprint sensor when I enter my PIN, and those combine to say, hey, he knows the right digits, and his fingerprint is the right one, this is the right person, if people really want to be just that extra level secure. Or fingerprint sensor along with face scanning. Why can't they both combine? And I feel like, now that I'm saying it, that we've talked about it before, and one of the points that maybe you mentioned is, if one of those things is out of whack, it makes it incredibly difficult for you to ever get into your device. Maybe that was you. I can't remember. Maybe it was someone else.

Steve: Yeah. In fact, I do remember you and I talking about that, and that's the other problem is that we've got multiple factors, and they're all a little bit soft. So you have, on one hand, you don't want a false positive, so you don't want to allow somebody in who should not get in. Nor do you want a false negative. You don't want to prevent the authorized user from getting in when they want to get in. I mean, that's very frustrating. I remember, I think it was the very first - was it the very first Touch ID? Something that Apple did just wasn't working that well when they first released it, and it was like [grunting]. I think it was the very first Touch ID. Or maybe I later figured out how to over train it, and I think that's what it was. I was able to over train it and then bring its reliability up where I wanted it to be.

But again, because it's a heuristic system that's inherently making a judgment call, does this look enough like the person I've seen before, because of course our faces are changing subtly over time. Our hands pick up damage from the environment, cuts and scrapes and things sometimes. So it's am I sure enough about who this is? But if it was a super high-entropy password, it's not a matter of being sure enough. It's like, holy crap, that's got to be the person because nobody else could get this...

JASON: You either know it or you don't.

Steve: Yes, could get this right.

JASON: Right.

Steve: And so we're dealing with the more soft decisions we make, the softer the overall compound decision becomes. And that means that we become more likely to false positive or false negative; and that doesn't help anybody, either. So really it seems very clear that biometrics is the future. People want the convenience. The only thing I would suggest to people, and we've said this on this podcast before, is give your phone to other people. Stick your phone in other people's faces. See if it unlocks for them. Have other people challenge your fingerprint reader. Develop some a priori real-world experience with how robust this is. I mean, I don't think enough people do that.

JASON: Most people don't.

Steve: Nobody asks someone else to, like, try guessing my password on my phone. I mean, that's just, you know, we know that's not going to happen. But it'd be really interesting to get some actual sense for whether you know other people whose faces will open your phone.

JASON: Yeah, absolutely. I completely agree. That's a very interesting exercise. I never thought to give my phone to someone, be like, hey, see if you can get into it.

Steve: We just assume. We just assume. We just assume.

JASON: And all that was needed was a $3 screen protector, and your fingerprint could open my phone, too. That's just creepy.

Steve: Yes, and in fact last April the Naked Security website reported that a Nokia 9's PureView fingerprint reader was fooled by a chewing gum packet.

JASON: But you had to know which chewing gum packet. [Crosstalk] security; right?

Steve: That's got to be Wrigley's double something.

JASON: Got to be Wrigley's, yes.

Steve: Yes.

JASON: Hey, before we go, I want to make sure that we get in here the chance to plug your SQRL event. It's coming up here in the studio; right?

Steve: Oh, yes, yes, yes, that's a good point, because Leo's not back before that, and we do want people to know. It'll be on Saturday, November 30th. I'm going to be up because I want to just make one final recording of my - basically the SQRL story is what I call it, sort of what happened, what occurred to me, what ensued, what SQRL is in a lot of detail. So it's not a user-facing presentation. It's meant to explain SQRL's technology for our audience, for the audience of this podcast. And so I've asked Leo if he would make the TWiT studio available for that purpose, and he and his staff have graciously agreed to do so. So that'll be the afternoon of November 30th, in about four weeks.

JASON: Awesome. Yeah, and I think I have down here 2:15 p.m. Pacific.

Steve: Yup, yup. Very cool.

JASON: I'm sure you'll hear a lot more about that. I'll make sure that we kind of get the plug in throughout the next month.

Steve: Yes. And it is by appointment only.

JASON: Okay.

Steve: So you absolutely - because there isn't that much room there.

JASON: Right, no.

Steve: And so you've got to, you know, if you're sure you can come, please don't make an appointment and then be a no-show because that would just be annoying to all the people who would like to come, but weren't able to. So if you're sure you can be there, then we'd love to have you.

JASON: Cool. Should they, I mean, normally we would send people to tickets@twit.tv.

Steve: Yes.

JASON: Okay.

Steve: I think it's already been created.

JASON: All right, great. So November 30th, SQRL. That's a Saturday, 2:15 p.m. Pacific. Email tickets@twit.tv if you are certain you can make it for that. That way we're not holding one of those seats from someone who could make it, and you end up not going, and everybody's sad. So don't do that.

Steve: Perfect. Perfect.

JASON: Don't make everybody sad. Steve, awesome stuff. What a great show, as always. People who want to check in on everything that Steve is up to can go to GRC.com. That's where you're going to find all the information about SpinRite, of course. You can get your copy there. Information about SQRL, if you want to find out a little bit more about what SQRL is all about leading up to November 30th, as well as audio and video of this show, can be found at GRC.com, and transcripts only found there. So if you want a transcript of this show and all other episodes of Security Now!, you can find them: GRC.com.

Our website is TWiT.tv/sn for Security Now!. There you can also of course find all of our podcast episodes, audio and video; subscribe links for every single place you might want to go. If you prefer YouTube or Pocket Casts or Apple Podcasts, wherever you get your podcasts, you'll find direct links out to those so you can subscribe very easily and have the show delivered to you automatically so you don't have to think about going to the site to play it from there. And you can watch us record the show live every Tuesday, 1:30 p.m. Pacific, 4:30 p.m. Eastern, 20:30 UTC. I don't think that's changed quite yet, but that's at TWiT.tv/live.

Steve, this was a lot of fun. Thank you for having me along on this ride. Really appreciate it. And we'll see you next week on Security Now!.

Steve: Thanks, Jason.