Contact Tracing Apps R.I.P.

Description: This week we begin with some browser news to examine a nifty new trick to be offered by the next Firefox 77, and spend a bunch of time on the many new features - and how to enable them - being offered in Chrome's 83rd edition. We also look at Adobe's four emergency out-of-cycle patches, and a surprisingly robust and well designed new jailbreak for iPhones. We take a look at a surprisingly powerful DNS amplification attack with a packet count multiplier of up to 1,620, the sad but true complete collapse of Bluetooth connection security, and the odd report of eBay scanning their users' PCs. We share a bit of closing-the-loop listener feedback and a quick bit of miscellany, then I editorialize a bit about why I'm very sure that contact tracking apps are dead on arrival.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-768.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-768-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here with updates on Firefox 77, Chrome 83, and why you can never trust your Bluetooth device again. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 768, recorded Tuesday, May 26th, 2020: Contact Tracing Apps R.I.P.

It's time for Security Now!, the show where we talk about your security, privacy, safety, health, and technology. I loved last week. Steve Gibson is here. He's the Explainer in Chief. He did a great job on WiFi 6 last week. I really appreciated that.

Steve Gibson: I got more feedback on that than I've had in a long time, Leo.

Leo: I loved it, yeah.

Steve: I was sort of actually a little self-conscious about did everyone want to have that. But the answer was yes, apparently.

Leo: You have a little schmutz on your chin.

Steve: I know, what the hell is that?

Leo: It looks like a mouse pointer.

Steve: I just tried to wipe it off the screen.

Leo: No, it's us. If I could get somebody in the main studio just to move that. Oh, don't eat it, no, it's - oh. That'll - can you delete - can you move the mouse, John or Burke? Oh. I don't know if there's anybody there. Hello. Oh, there it is. There it is. It's moved. Whew. That was close. We're in a funny mood today, I'll tell you.

Steve: Okay.

Leo: So yes, I enjoyed the WiFi 6 because you explain stuff so clearly and well, and I've always enjoyed those kind of computing fundamentals episodes we did way back when. So anytime you want to do that...

Steve: Well, and, yeah. And they're still there. But it's just not possible to get people to, like, we're all busy, and everyone wants to know what's going on. And there's a sense of, like, oh, no, we need new. So anyway.

Leo: We've got to do both. We've got to do both. You're the king of security news.

Steve: So we have Episode 768 today for May 26th, our last podcast of May. Where is the year going, Leo?

Leo: Wow, yup.

Steve: It's just zooming by. I didn't have a title for this initially.

Leo: By the way, it can continue to zoom, go by fast. Go, 2020. Not my favorite year. No, no.

Steve: Yeah. Unfortunately it's going to be a necessary part of solving the 2020 problem is living through it and acquiring herd immunity one way or the other. So I read something about some research that some modelers had done in the U.K. about the percentage of people who would have to be using contact tracing apps in order for the system to be effective. Thus the title, "Contact Tracing Apps R.I.P."

Leo: Uh-oh.

Steve: It's just - and that also gave me a segue, though, to sort of pull all this together. So we're going to end up talking about that a little bit. But we're also going to begin with some browser news to examine a nifty new trick to be offered by the next Firefox 77; and we're going to spend some time on the many new features, mostly on how to enable them, being offered in Chrome's 83rd edition, which just came out. And there's like a

whole bunch of cool stuff, but they're all turned off. So it's like, hello, okay, well, we want them, and they're all good, so we're going to turn them on.

Leo: Oh, good.

Steve: We've also got Adobe's four emergency out-of-cycle patches. We won't spend much time on that because, again, how can you have a remote code execution in something that animates characters? I don't know.

Leo: Oh, god. Oh, geez.

Steve: We've got a surprisingly robust and well-designed jailbreak for iPhones. I heard you guys talking about that on the previous podcast, on MacBreak. We take a look at a stunningly powerful DNS amplification attack with a packet count multiplier of up to 1,620, Leo.

Leo: Wow.

Steve: And in fact it's so powerful that it was responsibly disclosed, and it didn't come to light until most of the vendors had patched their DNS in order to resolve it. So we're talking about it now because it's not such a big problem. We've also got a startling event which, I mean, even as I say this, and as I was writing it earlier, I'm thinking, really? It's the sad but true complete collapse of Bluetooth connection security. It's just gone.

Leo: Oh, that's not good.

Steve: No, that's not good. Tell that to your front door lock and your garage door system and lord knows what else. Your security system. We also have an odd report of eBay scanning their users' PCs. Turns out not exactly what was happening, but sort of interesting. Then we've got a bit of closing-the-loop listener feedback; a quick little bit of miscellany about two shows I wanted to discuss with you quickly.

Leo: Oh, good.

Steve: And then I'm going to editorialize a bit about why I'm quite sure that contact tracking or tracing apps are DOA. It's just not going to happen. And as you already observed, we do have a pretty wonderful Picture of the Week.

Leo: Oh, it's good. It's very good. Steve?

Steve: So our Picture of the Week. I thought, you know, I wonder if this is on the 'Net anywhere. So I just, while you were talking, I brought up Google, and I typed in "Windows 10 we finally." And that's all it took. Because I was thinking it would really make great wallpaper. And anyway, for those who are not on the video feed, this Picture of the Week says "Windows 10: We finally fixed everything." And it just shows the most

screwed-up-looking passenger airliner. Actually, maybe not because there's no windows. I don't know what it is. But it's got like landing gear above the cockpit, facing the wrong direction. The wings are rotated 90 degrees, so instead of being at 3:00 o'clock and 9:00, they're at 12:00 and 6:00.

Anyway, it's quite wonderful. So if you're feeling like it's time to change your wallpaper, you just google "Windows 10 we finally," and it'll bring up plenty of hits of this because this turns out to be quite popular on the Internet.

Leo: It would go well on my Linux box, I think. It'll be very pretty.

Steve: Yeah, it's perfect.

Leo: Yeah.

Steve: A reminder of why you're...

Leo: On Linux.

Steve: ...having a much better time now.

Leo: Yeah, yeah.

Steve: So Firefox 77. We're currently at 76. It's going to pick up a nifty new trick. And I don't - well, okay. So one of the attributes of fill-in fields on an HTTP web form, any field can be given an attribute, a property called "maxlength." So a web page's coder can instruct the web browser to stop accepting user input beyond a specified length. It just, you know, if you set it to, like, five, for example, it would take five characters, and then it just won't take a sixth character, for example.

Anyway, supposedly, though this is something I don't ever recall encountering myself, there are websites that use this maxlength specification on their account creation and subsequent login password input fields. Okay, now, first of all, that's a very bad idea, but we'll get to why in a minute.

Leo: No kidding.

Steve: So as a result, when a user pastes a longer password, probably created by a good random password generator, into a shorter field, the browser, which has been clearly instructed to admit text no longer than "x" characters, will indiscriminately discard the overage without providing any user feedback. The website itself, the server at the other end, has no way of knowing that an attempt was made to paste more into the field since this truncation was all handled on the browser's end.

Now, as long as the site never changes its mind about the maximum length of passwords, and the user always goes about pasting the overly long password string into the too-short field, everything should be okay; right? Because the website would always

be receiving the same first "n" characters of a user's pasted password. But it's certainly conceivable that a site might, at some point, wish to modernize its authentication handling and, for example, increase its password length. For example, some random HIPAA or other regulation might say thou shalt allow passwords of up to "x" characters. And so to be compliant, the site might then be forced to widen its password length aperture to comply.

So now suddenly more of the user's same password would be admitted, and a great deal of breakage and hair-pulling would ensue. So we're currently, as I said, at Firefox 76. The next major release will bring an interesting new feature. I guess this is something that actually does happen to people, although as far as I know, I haven't ever encountered it, as I said. Since the web browser knows when we've attempted to enter or paste more characters into a field than that field has been configured to accept, starting with Firefox 77, when this is attempted, there will be an immediate bright red rectangular highlight with a message that says, "Please shorten this text to 'n' characters or less. You are currently using 'y' characters."

So I guess it wouldn't apply to inputting because you would be typing, and maybe you'd stop seeing it going in. On the other hand, password fields are often obfuscated with the big black dots. And if you kept typing, you might not know that they were no longer being accepted. So it'd be interesting to see exactly how Firefox handles that.

But anyway, it's just sort of a random weird thing that, again, poor site design. But on the other hand, there's no protection from poor site design. And I'll conclude, just by reimplanting what everyone in this audience very well knows. Password length limits are dumb. Period. Dumb dumb dumb. There's absolutely no conceivable justification for them.

Every recipient of the password, that is, someone who receives a password should immediately hash it; or, even better, in the browser, before it is even sent over the wire, script that is handling the form submission should pick a random salt, use it to drive a PBKDF (Password-Based Key Derivation Function), and then send the resulting hash with the salt over the wire so the recipient never has it in the clear. There's never any reason for any employee of the website, the recipient, to have or to see the password in the clear. They never have any need or reason to send it, or read it over the phone, or check it versus what, you know, in a conversation. The only proper way to deal with passwords that cannot be supplied for whatever reason is to have the user somehow reauthenticate their identity through some other means.

Unfortunately, and typically, the only way to do that is by having them able to receive email at a previously registered address, and then send the "password forgiveness link" to that email account. It's not great, but it's the way the world works today. So anyway, it makes sense for a password to be required to meet minimal complexity requirements which can be enforced by some script on the web browser. But there's just no good reason to place a cap on a password's maximum length or complexity. There just isn't. I can't think of a single good, I mean, yes, I get it that there are, like, backend mainframes at CompuServe still running Cobol that have a wired-in password length of 12. But in that case, take 12 digits from the hash and store that, rather than the password. Just there's no reason to limit what the user can do. So anyway, that's Firefox.

We just got Chrome 83 that we were talking about because remember that when the stay-at-home mandate landed we were at 80. And 81 was planned, but then was delayed. And because I guess Google is running on a calendar, they just said, well, we don't have time to do 82, so we're skipping it. But we're going to give everybody 83 on schedule. And so that's what happened. So we just jumped from 81, I had 81.0.4044.183, looks like an IP address all except for the 4044, that would not work, and

then went over to 83 dot whatever. And we got a bunch of new goodies. There were 38 security problems fixed, but mostly lots of UI improvements.

And I guess in the interest of rolling these out slowly, they're not enabled for anybody right now, which is - I don't get it. But at the same time I did have Chrome stop receiving Ctrl+Vs. I wasn't able to paste into Chrome after I turned all these on. I don't know that there's any connection between that happening. I think I've had that happen before in Chrome's DOCS app. And so I just closed Chrome and started up again, and then I was able to paste again. But anyway, that did happen.

So we have cookie management, both for Incognito and regular modes, simplified and clarified. We've got global and per-site website settings made more clear. The Site Settings control has been reorganized into two separate sections to make it easier and more clear. People item has been renamed to You and Google, which is where the sync controls are now located. And Google says that many people regularly delete their browsing history, which I didn't know about or of.

Anyway, they've moved Clear Browsing Data control for doing that to the top of the Privacy & Security setting, although I also noticed you can get to it even quicker through the More Tools pop-up menu off of the main menu. But any event, so Chrome also offers a new safety check feature which provides a number of services. It will tell its user if the passwords Chrome is storing have been compromised; and, if so, how to fix them. It will flag whether their own safe browsing technology is enabled and operating, or might have been turned off. You might have forgotten about it for some reason. It'll verify that the version of Chrome you're running is up to date and whether any malicious extensions are installed. It will tell you how and where to remove them.

The only problem, as I said, is with all of these things we're going to be talking about, it's not turned on. So you'll be going a lot to chrome://flags. That brings you to an unbelievably long list of settings. Now, Firefox has the same thing. It's useless, there are so many things there, without the search box. Fortunately there is one. So chrome://flags. And then in the search box put "privacy." What you'll find there is Privacy Settings Redesign, which is what they're calling it in this not-yet-released incarnation. So you want to switch it from default, which someday maybe default on - right now it's default off - and turn it to enable. Then it will, down in the lower right, a big relaunch thing will light up. You want to click that. It briefly shuts down and restarts Chrome. And then this new Safety Check option that wasn't there before appears.

I clicked Check Now, which is the big blue button, and I ran a check. I got all four checkmarks indicating that the things Chrome had just checked for me were all okay. So it doesn't feel like it's a big in-depth thing that it's doing. Maybe they'll add more stuff there in the future. They've got lots of room for it. So that would make sense. Also with this release of Chrome, Google has started blocking, and this was nice to see because Safari's the only browser that's ever done this, and I've never understood it, started blocking third-party cookies by default, not globally, but at least in Incognito mode. And because they wanted to make sure, I guess they feel this is an aggressive thing to do, there's a big banner you get in Incognito mode after once again you have turned it on.

So back at the chrome://flags page, this time you search for "improved cookie controls." That will find for you that item which you set to Enable, restart Chrome, and then go to the Incognito mode, which is all dark, assuming that your normal browsing is white, as mine still is, or light. And you get a little description of what Incognito mode is, followed by this banner: "Block third-party cookies. When on, sites can't use cookies that track you across the web." Sounds great. Let's have everybody turn on always for all browsers. But no.

Anyway: "Features on some sites may break," they say. So there's a switch there, and it's on by default. Anyway, new feature, thank you very much. I don't know if you have to enable the cookie control for that to be the case. I think you probably do because it's called - it's not just UI, it's Improved Cookie Controls. I don't know whether Incognito mode always had third-party cookie support turned off, or if it's on and now you can more easily see it.

There's also an Extensions Toolbar Menu. When enabled, it adds a little puzzle piece icon to the right of the group of extensions that you may already have displaying in Chrome, to create a quick-access dropdown for managing those extensions and their management. So again, Chrome flags page. And if you search for "extensions toolbar menu," you'll find it. Turn it on, restart Chrome, now you get the puzzle piece. And when you click on it, you get a cool little dropdown that allows you to more quickly access your extensions. Otherwise, you know, you go through the traditional menu, extensions, and then it takes you to a whole page. This is just kind of a little quickie.

Oh, and it did, in the sample that I have in the show notes, I was on a page that was not engaging any extensions. But I later tried it somewhere else, and it showed me which extensions had been activated by that page. That is, LastPass came up and said, yeah, I got some form fill stuff. And uBlock Origin, that's the other extension that I run, came up and said, oh, I just blocked 49 things. So it's like, thank you, uBlock. Anyway, cool little app. Of course, those things both show those items in their little widget tabs on the toolbar. So just another way of getting to extensions.

And the other UI improvement is the much-anticipated Tab Groups feature. Again, as with all these things, you've got to turn it on before you get it. So you want to search for "tab groups." That'll bring up three related tab group things. There's Tab Groups, which they say allows users to organize tabs into visually distinct groups, create separate tabs associated with different tasks. There's Tab Groups Collapse, which you have to separately enable, which allows the group to be collapsible and expandable. It's funny, because I tried that. And I put some tabs in a group, and then I said, yeah, collapse them. And everything went away. I mean, there wasn't anything left. There was no little hook to, like, get it back. And I thought, oh, okay.

Leo: It really collapsed.

Steve: It really collapsed...

Leo: It collapsed out of the universe.

Steve: ...the entire experience, yes, into the Tab Group black hole. Anyway, right-clicking somewhere on the bar gave me the option to get them back. And I thought, whew, okay, they're not gone forever. That's good. And they've done some nice things. They assign colors to the group. You can label the group. And then the tabs that are collected to the right of the group name, they have separate color highlighting. I mean, it's very pretty from a UI standpoint.

I don't use Chrome as my main browser. I'm still using Firefox with its wonderful vertical tab column that I just can't get away from. So I sort of use Chrome for smaller stuff. So I'm not needing to manage tabs. And I will say, sorry, Google, horizontal tabs are wrong. The moment people see Edge, where you can click on a little button in the upper left, and the tabs go vertical, it's the end of horizontal tabs across the galaxy. They will just cease to exist because they are fundamentally wrong.

Leo: You really like those, don't you.

Steve: Just wait. I mean, it's just like it's so much - it's obviously better. The tabs are - they take up horizontal space. Here's Google trying to shorten our URLs by removing arguably important things from them. But no, the tabs stay horizontal. It's like, okay, well, we'll see about that.

The good news is the Google UI person - no, no, no, I'm sorry, I'm confusing companies. It's Microsoft's Edge guy is in love with vertical tabs because he knows what's right, just inherently, intrinsically. Anyway, yeah, enough said. You can now Group Tab.

Leo: This theory is because we have wide screens, so there's more room on the left than there is on the top? Is that the idea? Why is it that you like these vertical tabs?

Steve: Well, Leo, when you have a hundred...

Leo: That's why. You can't fit them in this way. You can only fit them in that way.

Steve: And if you squeeze them, then you can't read them anymore.

Leo: You can't read them. It's just like one letter.

Steve: I mean, they're just wrong.

Leo: Yeah, yeah, yeah.

Steve: It's just wrong.

Leo: No, that makes sense.

Steve: To be across the top. It's so obvious.

Leo: Can you get a hundred vertically?

Steve: Well, I have a scroll bar.

Leo: Oh, wow. So you have so many tabs that you actually scroll them. Wow.

Steve: It's wonderful. It's just wonderful. Wait. The world will see. When Edge has that button, we're going to be on here. I'm going to say, "Now, Leo, I want you to reach up to

the upper left corner and press that little button that you've never pressed." And you're going to go [gasp]. And I'll go, "Uh-huh. That's what I'm talking about."

Leo: You know, I just don't keep that many tabs open. So it probably doesn't matter to me. I do pin tabs. I love pinned tabs. But that's only for stuff I don't change around a lot. I don't know.

Steve: I have some of those, too.

Leo: Yeah, I'm not a big - you're a tab guy.

Steve: Yeah. I'm kind of a messy desk person. So I've got this little tray along the top of my keyboard. And it's just got...

Leo: With paper clips. Old Lifesavers.

Steve: It just, likes, grows paraphernalia. Yeah. I've got like a little pushbutton, a little micro pushbutton.

Leo: Yeah, you might need that someday.

Steve: I've got an SD card for some reason, a microSD.

Leo: Okay, I get it now.

Steve: And I have a little surface mount speaker. And anyway...

Leo: But I have to point out, that tray is a horizontal tray. You wouldn't want a vertical one of those.

Steve: Ah, you got me. That's true. Although I do have my function keys on the left.

Leo: Oh, wow. Where did you get that keyboard?

Steve: My function keys are vertical, old-school.

Leo: You have an old IBM keyboard?

Steve: Northgate OmniKey 102 with the function keys where they're supposed to be.

Leo: Oh, man.

Steve: They are not supposed to be across the top. They're supposed to be running down there in two rows, or two columns on the left. And I lost that battle completely.

Leo: Yup. And the floppy disk is supposed to be right in the front there, the floppy. None of these little 1.5-inch whatever they are. The nice big 5.25-inch floppy. That's what we want, yeah. All right.

Steve: Lastly. The good news is on the Chrome flags you can type in "secure DNS" and enable Chrome's DNS over HTTPS, a.k.a. DoH. Remember that until it appeared in the UI, the way you had to do that was by adding a bunch of command line switches to the icon that launched your instance of Chrome. So this is way better than that. So that turns it on. And apparently it's going to work the same way as Windows 10 will when we get that DoH in Windows 10, which is if your system is set up to use a DNS provider that offers DoH, Chrome, when this is enabled, will just use DoH instead. So yay to that.

Oh, and as we know, back with Chrome, speaking of shortening the URL, with 79, that's when Chrome or Google pivoted back to deciding that displaying the http:// at the front of every URL is not useful. And more than that, that the "www" or the "m," which of course is for mobile subdomain, that you just don't need those. And as the person who did the write-up of this "bug," as it was called, said, they're better off being elided from the screen, so declaring that they were trivial subdomains.

With 83, as promised, those of us who are sticklers for details have reobtained the ability to display the full, unadulterated, unelided URL. We need to jump through a few hoops, but that's fine. You go to the chrome://flags and search for "omnibox context." That will find an item, "Context menu show full URLs." Got to turn it on because you wouldn't want to confuse people with an omnibox dropdown that didn't offer that option. Oh, my goodness, they might turn it on by mistake. So no, enable it first, then restart Chrome because, oh, have to do that. Then, yes, you can right-click in the URL field, and the last item there, mine now has a checkbox next to it, not surprisingly, that brought back my "www" and the https:// that has long been gone from Chrome.

Leo: Oh, thank goodness.

Steve: So thank you, Google.

Leo: Yes.

Steve: That really wasn't so hard; was it? No.

Leo: No.

Steve: Adobe surprised us last week with four out-of-cycle emergency updates. So if you were using Adobe's Character Animator, Premiere Pro, Audition, or Premiere Rush, they felt so strongly about these four vulnerabilities, one for each, that they broke with their normal monthly - and it was just a week before. They normally follow Microsoft with a

second Tuesday bumper crop, is what they had two weeks ago. They thought, oh, we'd better get this one out there. Because somehow their Character Animation app has a remote code execution flaw. And you've really got to wonder what in the world they're doing for a character animation tool to have a remote code execution vulnerability. But that's Adobe.

In any event, any of our listeners who are using them, any of those four apps, will want to check to make sure that they are updated with the current because they're all - one was remote code. The other was a worrisome information disclosure, as I recall. Or it might have been an elevation of privilege. I don't remember because I just got stuck on that how can character animation have remote code execution. But in any event, it does.

There's an iOS jailbreak that has just dropped. And of all the jailbreaks I've seen, this is the nicest and most professional that I've ever seen done. It really looks very nice. And of course it's time, this is a strictly time-limited offer because Apple's going to fix this any minute. So if you're moved to explore an iOS jailbreak and never have before, I would argue that this is the way to go, this one. And so it leverages a previously unknown zero-day flaw in, get this, every version of iOS from 11.0 through 13.5, which is where we are today, with two curious little exceptions. It doesn't work on 12.3 through 12.3.2, nor on 12.4.2 through 12.4.5. Those are excluded. Don't know why. But that means, since it goes from 11 to 13, it would work on the original iPhone 6s through today's latest iPhone 11 Pro Max.

And, I mean, it really has the feel of a professional jailbreak. I've never been tempted to do that. I heard you talking about this on MacBreak Weekly, Leo, and I agree with you. There's just, you know, yes, and I agree with Andy. It would be nice to have a better launcher. My biggest gripe with - I don't mean to diverge, but - with iOS is that I'll have so many apps that I don't know where they are. And I'll go through the pages looking for it.

Leo: Yeah.

Steve: And I can't find it. So I finally go, okay, fine.

Leo: I'll do the search.

Steve: So I pull down, go to search, type in a couple characters, and it finds it.

Leo: Most long-term...

Steve: But it just shows it to me...

Leo: I know.

Steve: ...in the menu.

Leo: It doesn't show where it is.

Steve: It doesn't help me know where it is. It's so annoying.

Leo: It used to show you the folder it was in. I know. They took that feature out. It used to say it's in this folder. And they took that out. I think that a lot of people just use the search mode. That's it. They just go, I don't know where it is. I'm going to pull it down. I actually organize everything into named folders, logical folders. But it's a terrible system.

Steve: And I saw yours, Leo. All of them had red things on them because...

Leo: Yeah, that's annoying, too.

Steve: ...inside every one of the folders...

Leo: Notifications.

Steve: ...there's something screaming for your attention.

Leo: Yeah. I have to go through everything and turn off all the notifications, which is a pain. Yeah. It's primitive. It's not much changed since 2007. But at the same time, here's my real question about jailbreaks. I mean, yeah, it's professional, all that. But a jailbreak always means they have to take advantage of a security flaw; right?

Steve: Yes. This is a zero-day flaw which they will - because, I mean, Apple doesn't want anybody to do this.

Leo: Right.

Steve: That's why it's a big deal. It's a jailbreak. If it was just a walk-in, then it wouldn't be a big deal.

Leo: Right, right.

Steve: Okay. So remember that CheckM8, also known as Checkrain, that jailbreak leveraged a flaw in earlier physical devices' boot ROM which allowed a single boot duration takeover of iOS. And that one worked on iPhones from 4s through iPhone 10. And because it was leveraging a flaw that was discovered in the boot ROM, those hardware platforms can never be fixed. Everybody will be able to use that who wants to on iPhones up from 4s up through 10. But not afterwards because Apple fixed the problem with iPhone 11. It's called the "Unc0ver," U-N-C-0-V-E-R. And anyone who's interested, unc0ver.dev is the site with 0 of cover being a numeric zero. I hope they grabbed the other one if it was available and bounced it over to the hacker spelling version.

So if you're curious to play with a jailbreak - and again, there's really not much you can do, as you said, other than download apps from non-Apple Store places, and that's of course fraught with risk. But the site explains that it's very compatible - they tested it on a gazillion devices - and very stable. They explain, they said: "Utilizing proper and deterministic techniques, jailbreak stability is guaranteed." They also claim security: "Utilizing native system sandbox exceptions, security remains intact while enabling access to jailbreak files." Meaning apps.

And under "Extensively Tested" they write: "Unc0ver has been extensively tested to ensure it's a seamless experience on all devices. Unc0ver works on all devices on iOS versions between 11.0 and 13.5. Below you can find a list of all devices that have been specifically tested." And, boy. When you click that "Show me the devices," it opens up a really long list of things that they've verified.

So anyway, they're very proud of this work. And they had something under "Important Information" that I thought it was important enough to share. So they explained that it's been "stable and enable freedom from the moment you jailbreak your device. Built-in runtime policy softener allows running code without Apple's notarization and pervasive restrictions. Proper runtime modifications to iOS kernel modify security features as necessary and result in..." and then they've got a number of things they're proud of. "No extra security vulnerabilities: Unc0ver preserves security layers designed to protect your personal information and your iOS device by adjusting them as necessary instead of removing them. With this security adjusted on your iOS device, you can run your favorite jailbreak apps and tweaks while still being protected from attackers.

"Stability and battery life: Unc0ver is tirelessly developed and rigorously tested with software stability and battery life in mind. If you're experiencing issues with stability or battery life, we recommend searching your device for faulty tweaks. Reconciliation of services: Services such as iCloud, iMessage, FaceTime, Apple Pay, Visual Voicemail, Weather, and Stocks have been reconciled and still work on the device." So other things don't break.

"Future software updates" - and this is a little confusing. They said: "The ability to apply future updates is retained. Modifications to iOS kernel are done in memory. This results in the jailbroken iPhone, iPad, or iPod Touch staying operable when a future Apple-supplied iOS update is installed." However, for iOS updates, they note: "Unc0ver Team strongly cautions against installing any iOS software update that breaks Unc0ver" - well, as the next one is sure to, 13.5.1 - "as you can't re-jailbreak on versions of iOS that are not supported" - yeah, there's a euphemism for you - "by Unc0ver at that time."

And then, finally, "Jailbreak legality: It is also important to note that iOS jailbreaking is exempt and legal under DMCA. Any installed jailbreak software can be uninstalled by re-jailbreaking with the restore root file system option to take Apple's service for an iPhone, iPad, or iPod Touch that was previously jailbroken."

So anyway, I'm very impressed. If you were ever thinking of playing with a jailbreak, I would argue this is the one. These guys really did a nice job of essentially doing nothing more than allowing you to use un, as they put it, notarized apps, unsigned apps, and only being able to purchase them through the Apple Store. So I'm not promoting it. But you can do the jailbreak under macOS or Windows. Technically you can use Linux, but you have to have an Apple developer account if you're going to use Linux. Windows and Mac have methods that don't require that.

And in related news, I'll note that the zero-day exploit broker whom we've referred to often, Zerodium, 14 days ago tweeted, on the 13th of May, that they would not be purchasing iOS remote code execution vulnerabilities for the next few months due to "a high number of submissions related to these vectors." They said, in other words, or I'm

saying in other words, there's apparently a bit of a market glut in iOS remote code execution. Or perhaps it's that hackers who've been stuck at home for the last few months have had more time on their hands to dig more deeply into iOS and are discovering a few additional gems there.

Okay. This is really interesting. Again, what happens when security researchers get very clever is the so-called NXNS attack. A group of cybersecurity researchers in Israel responsibly disclosed, back in December, to those who needed to know, details about their newly discovered way of using Internet domain name resolution system to hugely amplify, by up to a factor of 1,620 packets - meaning send one out, your victim gets hit with 1,620 in response - a DDoS attack to take down targeted websites.

We're learning of it only now because the many companies who are helping to run their portions of the Internet infrastructure, including PowerDNS, CZ.NIC, Cloudflare, Google, Amazon, Microsoft, Oracle's Dyn, VeriSign, and IBM's Quad9 have all since, they've already patched their software to address this problem. So basically, problem solved. But it was a really interesting hack.

So when a DNS lookup is requested, the request is almost always made, if not always made, to what's known as a "recursive DNS resolver." So like all of the DNS resolvers that we talk to, the ones our ISP provides, typically, to their clients. When you ask a DNS server for the IP address of a domain name, it goes to a so-called "recursive DNS resolver." Assuming that the DNS resolver does not already have the IP for the requested domain in its local cache, it will then take on the task on behalf of the user, while the user waits for a reply, of making the requests necessary to track down the IP.

And thus begins, in this case, the vulnerability. The resolving DNS resolver will first ask one of the top-level authoritative name servers for the IP of the name server that's authoritative for the second-level domain. So, for example, it will ask one of the many .com name servers for the name server that's authoritative for attacker.com. If the domain being looked up, say the domain being looked up is noodles.attacker.com, then having obtained the list of name servers from the name server for attacker.com that are authoritative for noodles.attacker.com, that - well, okay, yeah.

Leo: No, I like it. I'm enjoying it.

Steve: That recursive name server next asks the name server, one of the name servers that it's been told is authoritative for noodles.attacker.com, for the IP for noodles.attacker.com domain. And this is the problem. The attacker.com name server can be malicious. And it's easy to get, you know, a name server for a domain. Whatever domain you choose, you can get a name server for it. It can provide a long list of apparently unique, that is, distinct name servers, and there's no limit to how many name servers you can have. But in this case they all have the same IP, the IP you want to attack. At that point, the recursive name server that's trying to give you an answer will begin querying the victim IP, having been told that, yes, that that's where the name server is for the domain.

Leo: I'm going to check it again. Hello? Talking to you.

Steve: Exactly. That IP knows nothing about this. So when it either doesn't respond or it responds "Huh?" the user's recursive name server will then try the next fake name server in the list that it received from the malicious attacker.com name server. And because DNS runs over UDP, and packets can after all get lost, there's lots of retries involved.

Leo: Sure, yeah.

Steve: The result is a massive traffic amplification attack since there are also, after all, hundreds of thousands of recursive name servers located all over the Internet that are available to be queried to resolve the request. So the bad guy sets up the malicious name server at attacker.com, which disperses a long list of individual name servers with the IP of its victim. And then the attacker sprays a request for noodles.attacker.com...

Leo: All over.

Steve: ...to all of the recursive name servers it can find.

Leo: All over.

Steve: And all of them then launch a distributed denial of service attack against that single target victim IP. And basically it just melts. So it's a good thing they kept this to themselves because it would be a field day. The researchers said that the attack can amplify the number of packets exchanged, as I mentioned, by up to 1,620. They said: "Our initial goal was to investigate the efficiency of recursive resolvers and their behavior under different types of attack, and we ended up finding a new serious-looking vulnerability, the NXNS Attack."

They said: "The key ingredients of the new attack are, one, the ease with which one can own or control an authoritative name server; two, the usage of nonexistent domain names for name servers; and, three, the extra redundancy placed within the DNS structure to achieve fault tolerance and fast response time." They recommended that network admins who run their own DNS servers update their resolver software to the latest version. And interestingly, when I went looking for additional information, the www.nxnsattack.com site was unreachable, both for it and for the research PDF. It just timed out. So maybe they're getting a little of their own medicine at the moment, being DDoSed by someone who's at home and thinks that they wish that had not been solved or fixed. Who knows?

Leo: Sigh. That was a big sigh.

Steve: Well, yeah.

Leo: What's the matter, Steve?

Steve: Remember how we said that the Bluetooth pairing event is inherently insecure.

Leo: Yeah.

Steve: Because that's the one moment when two devices having no previous knowledge of one another are negotiating a shared key which they will henceforth share and use to re-recognize one another in the future.

Leo: Yeah.

Steve: And so we said, okay, so like go out into the middle of an empty parking lot if you were really concerned because the distance of Bluetooth is short. And maybe throw the tinfoil blanket over yourself if you really want to be careful, but probably not necessary.

Well, it turns out, as they say, that may have been necessary. But even that was not sufficient. What we have as a result of new research by a group who discovered a means of later performing exactly the sort of impersonation attack that the whole Bluetooth one-time pairing scheme was designed to prevent, is nothing less than a complete collapse of Bluetooth security. Their abstract of their detailed research paper says Bluetooth (BR/EDR), which we know is the standard Basic Rate/Enhanced Data Rate, essentially the standard communicating Bluetooth protocol, they say, "is a pervasive technology for wireless communication used by billions" - yes, billions - "of devices.

"The Bluetooth standard includes both a legacy authentication procedure and a secure authentication procedure, allowing devices to authenticate to each other using a long-term key. Both procedures are used during pairing and secure connection establishment to prevent impersonation attacks. In this paper, we show that the Bluetooth specification" - the specification. Again, this is not bugs. This is the spec.

Leo: This is how it's supposed to be.

Steve: This is how it's supposed to do it, kiddies - "contains vulnerabilities enabling impersonation attacks during secure connection reestablishment. Such vulnerabilities include the lack of mandatory mutual authentication, overly permissive role switching, and an authentication procedure downgrade to the legacy version. We describe each vulnerability in detail, and we exploit them to design, implement, and evaluate master and slave impersonation attacks on both the legacy authentication procedure and the secure authentication procedure.

"We refer to our attacks as Bluetooth Impersonation AttackS" - using "A" and "S" of AttackS, thus BIAS. "Our attacks are standards compliant and are therefore effective against" - yes.

Leo: That's a new one, "standards compliant."

Steve: "...any standards-compliant attack."

Leo: Oh, my god.

Steve: We broke no rules implementing this. Yes. And "therefore effective against any standards-compliant Bluetooth device..."

Leo: Oh, this is terrible.

Steve: "...regardless of the Bluetooth" - it's terrible, Leo - "Bluetooth version, the security mode, for example, Secure Connections, the device manufacturer, or the implementation details. Our attacks are stealthy because the Bluetooth standard does not require notifying end users about the outcome of an authentication procedure, or the lack of mutual authentication."

Leo: [Whimpering]

Steve: Right, it's quiet. "To confirm that the BIAS attacks are practical, we successfully conducted them against 31 Bluetooth devices incorporating 28 unique Bluetooth chips from major hardware and software vendors, implementing all the major Bluetooth versions, including Apple, Qualcomm, Intel, Cypress, Broadcom, Samsung, and CSR. So the BIAS attacks allow an attacker having knowledge of the Bluetooth address of either endpoint of a previously established connection" - that is, a pairing - "which is trivial to obtain in practice since it's being broadcast all the time, to successfully impersonate that device when reconnecting to the other endpoint."

So, as I said, we have nothing less than a complete and total collapse of Bluetooth's secure authentication. And what's important to understand, this is not, as I said, the result of any bug. It's a failure in the design of the Bluetooth standard. It's a disaster for Bluetooth security.

The researchers tested the attack against smartphones, tablets, laptops, headphones, and even single-board computers including the Raspberry Pi. Every device was found to be vulnerable to their BIAS attacks. The standards-setting body, Bluetooth SIG, said it's updating the Bluetooth Core Spec - yeah, no kidding - to "avoid a downgrade of secure connections to legacy encryption."

Here again we see the classic problem of security evolved, but required backwards compatibility. We saw this in all of the SSL specs over time. That was one of the early attacks we talked about was where you could pretend you didn't know about TLS v1.1. It's like, no, I don't know. And so it's like, oh, well, we'll still connect to you. And wham, security compromise.

So Bluetooth has the same problem. They allow a connection that was previously established over a secure connection to not remember that that's the way it was originally connected and thus the devices can connect securely. There's no flag set for that. There is in the revised spec, but right now it means it's not there, and so any device can claim ignorance of a secure connection and ask for a legacy less non-secure connection, legacy encryption, in order to get a weaker link. Which allows the attacker then to initiate a master-slave role switch, placing itself into the master role and becoming the authentication initiator, which then allows it to leverage a couple other problems with the spec.

So in addition to urging companies to apply the necessary patches - there are patches, either in existence or coming. The Bluetooth SIG is recommending Bluetooth users install the latest updates from device and operating system manufacturers. So the good news is a lot of this can be fixed in the Bluetooth stack, which can be updated in the field. The bad news is how many devices are never going to be updated, like the Bluetooth-enabled front door lock that many people have. Or their Bluetooth-based security system that already has other updates we've talked about that are never going to be fixed.

Anyway, the research team concluded: "The BIAS attacks are the first, uncovering issues related to Bluetooth's secure connection establishment authentication procedures, adversarial role switches, and Secure Connections downgrades. The BIAS attacks are stealthy, as Bluetooth secure connection establishment does not require user interaction." And all of us who use Bluetooth know that. We're not being annoyed, which is what we would consider it, if every time our pencil gets within the tablet range, something happens. No. You do the pairing once. They learn about each other. They remember each other. And then they simply reconnect when they're within range. Well, unfortunately, if you reconnect with a spoofed other device, nothing tells you. So it's patchable. But how many devices are ever going to get patched?

Anyway, important security research, but not good news for Bluetooth. Hopefully we will see updates for all of our, you know, the OS-supported stack and updates in our devices in order to fix this. Again, it's not obviously clear what it means for the individual user. But it's, again, the sort of thing that would just have law enforcement salivating because they're probably thinking, oh, I know exactly how we can use this.

Leo: Is it theoretically possible you could create a device that would just walk up to a Bluetooth-enabled door lock and open up the door? Say, yes, I'm Stacy's cell phone. Hello.

Steve: No. Well...

Leo: Because there's a PIN; right?

Steve: That device needs to have been near Stacy's cell phone in order to get her Bluetooth address. Then it's able to turn around and pretend to be her cell phone without knowledge of the key. That's the deal is that it's able to say, I'm Stacy's cell phone. I don't support the high level of authentication. We need to use legacy.

Leo: Right, right.

Steve: And I'm going to be the master in this reconnection establishment. The other end says, oh, okay. Hi, Stacy. And then it pretends to impersonate without needing to know what the key is.

Leo: So this is kind of really on the order of something that a nation-state would use.

Steve: Yeah.

Leo: You'd have to be a target. I'd have to say, okay, I want to get in that house. I know it's got this kind of lock. Let me do some sort of "Bourne Identity" rendezvous with Stacy to get her cell phone cloned.

Steve: Exactly.

Leo: Yeah, yeah, okay.

Steve: Yup. So in a weird piece of news, the headline that I saw was "eBay port scans visitors' computers for remote access programs." And I thought, huh? Okay, but it turns out that's not really what's going on. Kind of. What is going on is - maybe it's clever. I don't know. It doesn't really bother me. It bothered Lawrence Abrams at BleepingComputer, who wrote about this. So what's going on is when a user goes to eBay's website to bring up the eBay page, in the process some JavaScript named "check.js" runs on their own browser to internally probe 14 specific ports on their PC at 127.0.0.1, the localhost IP. It uses the WebSocket protocol that allows that to see whether any of that set of 14 well-known remote control apps - actually it's fewer apps and more ports. There's, like, four for VNC, four for TeamViewer. But there's RDP port 3389 is checked. Four for TeamViewer. Anyplace Control and AnyDesk and AeroAdmin. Anyway, it checks for them.

Lawrence, who covered this issue, as I mentioned, on BleepingComputer, he wrote: "As the port scan is only looking for Windows remote access programs, it's most likely being done to check for compromised computers used to make fraudulent eBay purchases. Back in 2016, reports were flooding in that people's computers were being taken over through TeamViewer and used to make fraudulent purchases on eBay. As many eBay users use cookies to automatically log into the site, the attackers, who were able to remote control the computer, were able to access eBay" - that's kind of clever, actually - "to make purchases."

He says: "It got so bad that one person created a spreadsheet to keep track of all the reported attacks, and many of them referenced eBay." So he says: "The script being used for fraud detection is further confirmed by Dan Nemec's great write-up, where he traced it to a fraud detection product owned by LexisNexis called ThreatMetrix. As part of ThreatMetrix's description, they discuss how they detect and protect sites from Remote Access Trojans."

ThreatMetrix's product page explains: "Malware protection helps businesses mitigate the risk by being protected from Man-In-The-Browser (MITB), Remote Access Trojan, high velocity and high frequency bot attacks to low and slow attacks mimicking legitimate customer behavior, ransomware, key logging attempts, et cetera."

So he says: "While the programs being scanned are all legitimate, some of them have been used as Remote Access Trojans in phishing campaigns." And he concludes: "Regardless of the reasons, port scans like this are intrusive and not something that many users would want to have happen when visiting a site." And it's like, well, okay, maybe. It sounds like it's for the user's own good. I don't know what happens if you have a server running. Remember that that doesn't mean that that port which your browser is able to see open is open publicly because it would have to be mapped through a NAT router, which hopefully it's not. Or maybe TeamViewer uses Universal Plug and Play to open incoming ports to itself. Who knows? Anyway, just thought that was interesting.

Leo: I think it's interesting that JavaScript will tell you that. I mean...

Steve: Yeah. JavaScript, well, for example, SQRL uses that technique to talk to the SQRL client which is installed in your computer. So SQRL, the login page connects to the SQRL client on the localhost IP in order to get it to pop up the dialogue asking the user to verify their identity. And then the client performs the negotiation and provides a token back to the web browser, which immediately cuts out any man in the middle. So in this

instance we were using it to create very strong security. And Microsoft has talked about cutting browsers off from having local posts or - I've just forgotten the word. Local...

Leo: Port?

Steve: 127.0.0.1.

Leo: Oh. IP address? Localhost.

Steve: Localhost.

Leo: Okay.

Steve: Yeah. I was just blanking on local. So cutting browsers off from having access to the user's own local stack. However, when they tried that, it broke so many things. It turns out it's very useful to be able to...

Leo: I can see why you'd need that and be able to query a specific port and say, yeah, okay. But it just - this is the thing to remember is that these browsers run software, and the software has a lot of capabilities.

Steve: Yeah. Yup.

Leo: You know? And this can be...

Steve: And we're deliberately giving them more every day because we want them to turn into little app containers.

Leo: Yeah, that's right.

Steve: And actually be apps. So I got a tweet from Igor Lima, @igorlimatweets. And I mentioned this. He said: "Loved the WiFi history, Steve. Really appreciate the detail you provided, especially MIMO beam forming and collision detection. Please continue providing such historical context in future episodes." And that was sort of a placeholder for me to mention, you already did at the top of the show, Leo, but I got a lot of feedback from who appreciated that. So I will certainly take that under advisement.

Brian Helman tweeted. He said: "I listened to the latest Security Now! today. Three comments: I didn't know there was an 802.11 [period, no letter] wireless implementation. I always thought 802.11a was first. Doesn't that make 11ax v7, though?" And so he now goes 11a and then b, then g, then n and ac, then...

Leo: Starts counting. Looks like "b" preceded "a," which is really confusing.

Steve: Yeah.

Leo: And that 11 nothing. But that was so slow, that was 1Mb. No one used it.

Steve: Oh, yes, yes. It's like, okay. Why am I using my radio?

Leo: Yeah.

Steve: Anyway, so he said - I mean, he's right. But, you know, now we have six. He says: "Second is I'd have loved to hear why they picked the names the way they did instead of 11, 11a, b, c." And who know, engineers, IEEE, go figure.

Leo: And why was it b/a/g?

Steve: Yeah exactly. And what happened...

Leo: Right? It wasn't alphabetic.

Steve: ...to d, e, and f?

Leo: Yeah, I know, it was just random.

Steve: Yeah. He says: "Lastly, you left out the biggest advancement with 11ax, unless I missed it, that we don't use because IoT manufacturers lag so far behind, and that's OFDMA (Orthogonal Frequency Division Multiple Access), allowing sub-channelization to reduce data rates to sub-2Mb." He says: "This is HUGE," all caps. "It keeps low-bandwidth devices from hogging full channels, freeing up space for devices that need the bandwidth." And he's correct. I did talk about how with 11ax there were 2,000 separate sub-channels.

I did not talk about what that means is that the access point is able to divide all the bandwidth up to individual clients to prevent client collision. I talked about that aspect of it. But it also means, because each of these sub-channels is very low bandwidth, there are all kinds of IoT devices that don't need the full triple-scoop 80Mb bandwidth and are quite happy to just trickle out data at 1Mb. And so "ax," when our devices all support it - and of course that's the problem is they have to be ax-aware in order to negotiate that. But in the future they'll be able to just ask for a little sipping straw instead of the big fire hose. And our systems will work much better as a result.

Liron Amitzi, hope I did that right. He says: "Listening to you talk about DoH and wondering, if ISPs start having their own DoH-enabled DNS, wouldn't they still be able to monitor us? Even if the transport is encrypted, they own the DNS servers. Am I missing something?"

No. That is right. And it's why I expect that before long ISPs will be running their own DoH. They're going to want not to be cut out from that traffic, it's very clear. But the privacy-concerned user has the choice to say, no, I'm going to use Cloudflare or Quad9

or whatever, and continue routing their DoH traffic to the DNS server of their choosing. And someone tweeting as Classy Gay INFJ, he said: "@SGgrc Hi, Steve. Can you tell me what this device is? It looks very familiar. Thanks."

Leo: I know what that is.

Steve: He's captured a screenshot from the podcast over my left shoulder. You can see it right there in the background, to the left and below the three PDP-8 clones. That thing is near and dear to my heart because I helped develop it.

Leo: Oh, I didn't know that.

Steve: That's a Texas Instruments Speak & Spell.

Leo: I had no idea. I mean, I knew that it was a Speak & Spell.

Steve: And that was the first device that was generating synthetic speech. It used a linear predictive coding codec in order to compress sounds. Back then, 4Kbits was a big ROM, and actually 4Kbits is what that thing had. And despite the fact that it was incredibly lean in memory, it was able to speak. And I had a hand in making that technology happen. So anyway, that's why that's there behind me. And it's funny, too, because I just ran out of time. It works, and I meant to have it so I could - my very favorite thing was it used to - one of the words that it asked you to spell, because it's a little tricky spelling-wise, was "relieve." And of course so it would say "Spell relieve." And then I would type R-O-L-A-I-D-S.

Leo: You'd have to be of a certain age to recognize that reference, I think.

Steve: That's right. And, finally, Stuart Donaldson.

Leo: Wait a minute. Now, as long as we're asking, because you're just going to get another question, so what's this thing right here? It's next to the tape, the computer tape.

Steve: Ah, yes. That is a field maintenance panel for a hard disk drive. It's an exerciser. Back in the day you needed to exercise hard drives, you know, when they were the size of washing machines. And so a field service tech would unplug the computer and plug this thing in, and it pretended to be the computer. So it would issue seek commands, and he was able to, like, put the thing through a controlled testing sequence.

Leo: And I love the magnetic tape next to it from an IBM or some such mainframe. And what is that, a DAT recorder above your head there? Next to the OEDs? To the right of the OEDs? It looks like a VCR with a keypad on it. Here, let me show you.

Steve: Oh, oh. That's actually the front panel of an Interdata 1116 or 11 - it's another minicomputer...

Leo: Your office is full of more junk than mine is.

Steve: ...from my history, yeah, yeah.

Leo: And I recognize the real PDP over there. And then I don't know what those - I'm just trying to save you more tweets, more things to respond to. I love it.

Steve: All my gadgets, all my digits...

Leo: There's a lot going on in this background there.

Steve: That's right.

Leo: [Crosstalk] much attention to it.

Steve: I'm not sure that that's a good thing. I've noticed that everybody is now doing podcasts and...

Leo: Lots of backgrounds, too.

Steve: Yes, things from home. Everyone is like, oh, look at those. Look at that tree, it's - oh, look at that globe. And it's like, okay, well, a little distracting.

Leo: There's a whole Twitter account called @ratemyskyperoom, where there's rating people's backgrounds.

Steve: Oh, my lord. Anyway, to finish up, Stuart Donaldson said: "Hey, Steve. You messed me up. You turned me on to Peter F. Hamilton."

Leo: Oh, good.

Steve: He says: "I just finished up the audiobook 'Naked God,' the last book in the Nights Dawn Trilogy." And weren't there five books in the trilogy? I think there were more than just three.

Leo: Oh, I don't know. It was a long...

Steve: Or maybe I'm thinking of [crosstalk].

Leo: It was a trilogy...

Steve: But definitely...

Leo: There were a lot of books.

Steve: Like the fourth book in the Hitchhiker's Guide trilogy. Like, okay, Douglas. He says: "Now I have six weeks of Security Now! to catch up on." Oh, because he was listening to the audiobook, and Peter Hamilton never saw a book that he couldn't make longer. So anyway, he says, "Stay safe." And I wrote back, and I said, "My recommendation for your next read, Stuart, is 'Fallen Dragon.' It's fabulous, and it's a rare Hamilton stand-alone novel. Then you must read 'Pandora's Star' and 'Judas Unchained,'" which are two books.

Leo: And I think we both agree that "Fallen Dragon" is the best starter Hamilton novel because it's a single novel. It's incredible. And all of the elements of his fiction are in that one book. It's really good.

Steve: Yeah, yeah, really good. So in Miscellany, two little bits, Leo. I wanted to just mention "Bosch" on Prime.

Leo: Yeah, oh, yeah, we've watched all of it. We're going to watch the new season sooner or later. Yeah, I like it. It's good.

Steve: Okay. Many people recommended it. I finally started. And I'm in the beginning of the fourth season and just absolutely loving it. I think it's just a pitch-perfect hardened police detective series. And I didn't know there was going to be another season, so I'm delighted.

Leo: Yeah, I didn't either, yeah.

Steve: And speaking of another season, HBO is doing "Perry Mason."

Leo: I saw the trailer for this, and I'm getting - that's exciting. It's the origin story for Perry Mason.

Steve: Yes, yes. Robert Downey, Jr. was going to play Perry, and I'm glad he had a movie-making conflict because I would not have liked him as Perry Mason. I don't think I would have been able to buy him playing that part. But Matthew Rhys, or Rhys...

Leo: I think it's Rhys. He was Henry VIII in "The Tudors." And I loved - is that who it was?

Steve: He's a fabulous actor.

Leo: He's really good.

Steve: He was also the husband in "The Americans."

Leo: Yes, yes.

Steve: Yeah. And...

Leo: I'm thinking of Jonathan Rhys who's Henry VIII.

Steve: Oh, okay. Yeah, Matthew Rhys was the husband opposite - I can't think of her name now.

Leo: Oh, I love her, too, Keri...

Steve: Keri, yeah.

Leo: Keri, yeah. We all know her.

Steve: But also in "Perry Mason" - by the way, it's coming to HBO June 22nd - we get Tatiana Maslany is back. And of course we talked about her on this podcast because she played, what, nine parts in - now I forgot that one.

Leo: Who was she? Is that "Russian Doll"? No.

Steve: No. Remember - was it "Black Mirror"? No, not "Black Mirror." That's a series. Somebody will know. Tatiana Maslany. Anyway, we were just blown away that she was able to play so many different...

Leo: Oh, "Orphan Black."

Steve: "Orphan Black."

Leo: And she was all the clones of herself, yes.

Steve: Yes, yes, yes. Thank you.

Leo: She's the new Della. No, she's Sister Alice.

Steve: So my take on contact tracing apps being DOA.

Leo: Do you want me to do an ad before you do that? We have one more.

Steve: Oh, I didn't know we had one more. Yes, yes.

Leo: Oh, you're replete with advertising today. You have the full set, which is a very good thing.

Steve: That last one was the penultimate ad.

Leo: Oh, dragging out the big words. Let's talk about contact tracing apps, Steve Gibson.

Steve: So I believe that it's doomed. As I mentioned at the top of the show, some academics who have modeled the system have determined that, to be effective, 80%, eight zero percent of all smartphone users would need to voluntarily opt into using the app. And that's never going to happen, 80%. As we saw, the instant the Apple/Google initiative was announced, both the nontechnical and, sadly, the technical press went berserk over the privacy implications. Even highly technical individuals who should have known better spoke out with errant, frightening, and unfounded warnings before they understood how the system worked.

This podcast looked at the systems technology carefully and understood exactly what and why the Apple/Google team had designed it as they did. And we found that the API itself absolutely protects the user's privacy. But in practice, that doesn't matter at all. That is, the fact that you really can trust it. For one thing, as we've discussed since then, health officials really do have a need to collect real-time geographical location data as part of a workable system. Adding the "where you were when it happened" would go a long way toward making up for a lack of pervasive use of an application.

For example, if only a few people in a large gathering were app-enabled, and it was determined from the app that that was the most likely infection event, then a call could be put out for other non-app-enabled people who were also present at that event to take the necessary precautions. As we noted previously, the importance of knowing where, which Apple and Google scrupulously avoided using, has already occurred to the state of Utah, who has created a much more useful solution, which is also necessarily much more invasive, even though it was thoughtfully designed with things like immediate user deletion of all data and short-term self-expiring location data.

The simple truth is a short-term sacrifice of privacy is required for spreading events to be located and managed. Even fully human-mediated contact tracing is by definition a short-term sacrifice of privacy. Someone whom you've never met and don't know anything about needs to interview you to determine everything you're willing to share about where you've been, what you've done, and who you've been in contact with for the previous two weeks.

Leo: And that better be pretty complete, yeah.

Steve: Yeah.

Leo: Yeah, don't lie.

Steve: That's a massive imposition on one's privacy.

Leo: Right.

Steve: But it's what's necessary.

Leo: It's what works.

Steve: And I learned in doing the research about one of those new features in Google, Google says that people are constantly clearing their web browser histories, and that's just cyber. Many people apparently really don't want anyone else to know where they've been and what they've been up to. And at least when interviewed by a human contact tracer, someone can choose to elide anything they're embarrassed to share. But you can't do that with an app. So how many people are going to voluntarily install what amounts to spyware? As we well know, many people also have an inherent mistrust of the government and its motives.

We've already seen people worrying that this might just be the start of more pervasive monitoring, I mean, even Andy on MacBreak Weekly was worrying about that, the start of more pervasive monitoring, with statements like - and I'm not quoting Andy, I had this already written - "If they are allowed to do this, they'll always want more, and why would they ever want to stop?" and so forth. So no. It was a noble idea. I loved the cleverness of the technology. It was fun to dissect it and share its operation with our users. But it's clear that as a voluntary initiative it's never going to get off the ground.

So I just, you know, to me, the idea of states using a more invasive technology, as I noted, really does make up for the fact, it goes a long way toward making up for the fact that just not that many people are going to take advantage of it. But for those who do, who are involved in events where an infection is present, if you know where and when, then you're able to move that from cyber into physical. And after all, that's what human contact tracing is, is physical contact monitoring. So I just think there's just no chance for software-based apps, if they are voluntarily installed.

Leo: And Bruce Schneier brought up the really excellent point about - and I don't know if you read Schneier's post a few days ago.

Steve: I did not.

Leo: He agrees with you. He says it's just - it's not going to - it's just they don't work because of the potential for false negatives or false positives. People aren't going to trust it. And it's just not going to work. We need human contact tracing. Now, I think these apps could be a useful adjunct to help with - because one thing, I don't remember where I've been, everywhere I've been in the last 14 days.

Steve: Yeah, good point.

Leo: Having a map of that, that maybe only I see or whatever, that would be useful.

Steve: It would jog your memory.

Leo: Yeah. There's a lot of things that could be useful. But as constituted, Apple and Google have done such a good job of protecting our privacy, they've made their apps pretty much useless to human contact tracers. And we know that human contact tracing is - you need extensive testing and tracing, period. So, yeah. Unfortunately...

Steve: And I think maybe it suffers a little bit from - I think the Apple/Google approach maybe suffers a little bit from just being developed by techies without contact from actual health officials, who would have said, wait.

Leo: We need that.

Steve: We have to know where this happens.

Leo: It's not useful. No, that's what Schneier said. He said tech will always respond with what tech knows how to do because they want to help. But that's not necessarily what needs to be done. It's just what they can do. And I think that's, I mean, I agree, they did it very well. But it's just, from a health perspective, health official perspective, it's not sufficient.

Steve: I would love, as I mentioned when we talked about Utah, I would love for the state of California to do an invasive health monitoring app. And I would load it.

Leo: Well, they're not using the Google API. They're going to do their own thing. The problem in my opinion is, I agree, we need testing, and we need human tracing. The countries that that's been done in, it has worked. It's worked very well. It's eliminated COVID-19 cases.

Steve: Just killed it.

Leo: Just killed it. But I don't think it's ever going to happen in the U.S. We just - we're not going to sit still for that. We won't even wear masks.

Steve: I agree. I agree.

Leo: This is a country founded on individual liberty, and it's inconsistent with what is necessary to stop the virus. That's why there's all the emphasis on, well, let's get a vaccine, because they know there's nothing else that's going to work.

Steve: Right.

Leo: And what that means is many, many, many more people will die. We'll have many more, you know, go back inside. It's not over. And then coming out, and going back. And it's going to be a bit of a seesaw for a little while, yeah.

Steve: I think that's right.

Leo: Well, there you have it. Do recommend both the Schneier article on it, and then he refers to a Brookings Institute article which came up with the same conclusion that you have, as well, independently. So, yeah, I think it's...

Steve: And you had some information that you mentioned on MacBreak Weekly about what states are doing. Are states going their own way?

Leo: Yeah. So only four states have so far said that they are going to make an app supporting the API, the Apple/Google API. Alabama, North Dakota, I can't remember the two others. There's a list...

Steve: Oh, my god, those two need invasive contact tracing more than anybody else. Alabama's in horrible shape right now.

Leo: That's right, yeah, exactly. But California, in fact it was an article - Mikah Sargent referred to it in our show earlier today, iOS Today. I wonder if I can find it. But it was an article by, I think it was - oh, it was Zak Hall, I think, as I remember, who actually went to the trouble of calling every state, every state health department, and trying to figure out which states were going to do what. And he has an actual list, state by state list, and quotes from each state health director about the plans.

And, yeah, only four states have any plans to, at this point, anyway. And Latvia. Latvia's going to use the Apple/Google API. So it's not exactly widespread acceptance. It's a tough one we're in, Steve. But I'm glad we're quarantining together, you and me, you with your Speak & Spell. Didn't that feature in an early episode of "Halt and Catch Fire"? Didn't he - that's how he got into making these computers? They took apart a Speak & Spell? Except they didn't call it a Speak & Spell because...

Steve: You know, one place I do remember it, I think that ET used it as part of his...

Leo: ET did, as well, yeah.

Steve: Yeah.

Leo: Yeah, Richard Dreyfuss, yeah. Wow. See, we're standing in the presence of giants, ladies and gentlemen. Steve Gibson.

Steve: Well, we're sitting, at least.

Leo: We're sitting. I'm sitting on the ball, but we're sitting in the presence of giants. Steve Gibson's at GRC.com. That's where you'll find, not only his great SpinRite, the world's best hard drive recovery and maintenance utility, the only thing he charges for, lots of free stuff, lots of informational stuff, ShieldsUP! and so forth. Plus of course you'll find this show. He's the only person in the world who has 16Kb versions of this show for the bandwidth impaired. He also is the only person with full transcripts, written by Elaine Farris, so they're very good, very accurate. So if you like to read along while you're listening, or maybe you like to just read instead of listen, that's all there at GRC.com. He also has 64Kb audio.

We have the audio, but also the video at TWiT.tv/sn. We do the show on Wednesdays, I'm sorry, Tuesdays, 1:30 Pacific, 4:30 Eastern. That's 20:30 UTC. If you want to watch, tune in Tuesday. You can just go to TWiT.tv/live. There's multiple streams there, audio and video. After the fact it's easy enough. You just go to our website or Steve's website and download a copy. Or, and this in my opinion is the best way to do it, get a podcast application, there's plenty of them, and subscribe. That way you'll get it automatically. You probably want every episode. What is this, what did you say, 768?

Steve: 768.

Leo: Wow. I think you should really have the complete set. Then you can follow the ebb and flow, 15 years or something of security. Steve, thank you, have a great week, and we'll see you next time on Security Now!.

Steve: In June.

Leo: Wow. How did that happen? Halfway through.

Steve: Bye.