Promiscuous Cookies

Description: This week we offer some welcome news about Microsoft AV under Windows 7, we follow even more blow-by-blow consequences of January's final updates for Windows 7, we look at a worrisome exploitable Bluetooth bug Google just fixed in Android and what it means for those not fixed, we update on the Clearview AI face scanning saga, we take a peek into data recovery from physically destroyed phones, we entertain yet another wacky data exfiltration channel, and we conclude by looking at the consequences of the recent changes to make cookies less promiscuous.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-753.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-753-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We've got lots to talk about, including the return of Security Essentials to Windows 7, plus two new bugs. Hey, no big deal. We'll also talk about the surprising revelation that the CIA has been spying on everybody, every customer of Crypto AG for 40, 50 years. That, and we'll talk a lot about Google's plan to eliminate third-party cookies. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 753 recorded Tuesday, February 11th, 2020: Promiscuous Cookies.

It's time for Security Now!, the show where we cover the latest insecurity news, cover your privacy, cover your privates, and we give you all the information you need to protect yourself with this guy right here doing the Vulcan salute. Ladies and gentlemen, I give you Mr. Steve Gibson. Hello, Steve.

Steve Gibson: You've got to watch your privates this episode, Leo, because we're going to be talking about promiscuous cookies.

Leo: Oh, my goodness.

Steve: And so you want to make sure you don't get any cookie crumbs.

Leo: Yes. Well, it is a privacy issue; isn't it.

Steve: Yeah, yeah.

Leo: All right.

Steve: So we have, we're going to start the week with some welcome news and an apology that I have to give to my Twitter followers, since they must have been tweeting at me about this. But I've been so busy and focused on other things that I've fallen behind in keeping up with Twitter because it was Elaine who first provided some good news, which we will discuss, that I'm sure other people were trying to tell me about. And I'll hold us in suspense for a moment about that.

Then we're going to follow even more blow-by-blow consequences of January's final updates to Windows 7. They're just really having a problem with this last one. We look at a worrisome exploitable Bluetooth bug Google just fixed in Android. But unfortunately, it's potentially bad, and we know that a lot of older Android smartphones aren't going to get fixed. So we're going to take a look at that. We're also going to update on a subject from last week, this Clearview AI company, and the ongoing saga with them as more major companies have awoken to the fact that they've had their sites scraped and have not taken it well.

We're going to take a peek into data recovery from physically destroyed phones. Our NIST revealed some interesting work on the problem of getting data off of phones which bad guys have attempted to physically destroy in order to prevent that recovery. We look at yet another wacky data exfiltration channel. And then we're going to conclude, as I mentioned, by looking at the consequences of the recent changes to make cookies less promiscuous for this Episode 753 for February 11th. So I think another great podcast for our listeners.

Leo: Nice.

Steve: And a very techie, geeky Picture of the Week, but something that I've had in my pile, waiting to share, that will be fun. So, yeah.

Leo: I have a Video of the Week to share for you.

Steve: Okay.

Leo: So when you do the picture, I'll do the video. How about that? It's a little surprise for you, Steve. But first - well, no. It's a pleasant surprise.

Steve: Yeah, I knew you weren't going...

Leo: I wouldn't do anything mean. All right, Steve. Who first? You want me first or you first? You want the picture or the video?

Steve: Let's see the video. What have you got for us?

Leo: We had our engineering dinner last night. And Patrick Delahanty, I don't know if you know, he's our guy, he does all our programming, he manages the API. He's really our coder, our security guy. He moved back East with his wife Svet, who is a children's author, very successful. And they have a beautiful boy, [Kaden], who is - how old is [Kaden] now, maybe a year and a half? Maybe just a year. He sent us this video of [Kaden]. Apparently [Kaden] has excellent taste in podcasts.

[BEGIN VIDEO]

ADULT: Are you watching Steve Gibson? Steve Gibson.

CHILD: [Laughing]

ADULT: Steve Gibson.

CHILD: [Laughing]

ADULT: Steve Gibson.

CHILD: [Babbling]

ADULT: Yeah.

CHILD: Ha.

[END VIDEO]

Leo: He loves Steve Gibson. He asks for Steve Gibson. This guy's going far. I think he's a red teamer. I think. I don't know. I may be wrong. But this kid [Kaden] is adorable. And he loves, for some reason...

Steve: Yes, the Mr. Rogers of security.

Leo: Even children love Steve Gibson. Isn't that awesome? All right. Now your picture. Now your turn.

Steve: Well, so all of us, I'm sure, you and I have discussed this particular breed...

Leo: Oh, I hate CAPTCHAs.

Steve: Yes, this particular breed of CAPTCHA is so annoying. They take a photo and then chop it up into a grid. And, you know, typically it'll be like an intersection with various things going on, and your job as the human is to select all the squares of the grid that have an auto, or that have a crosswalk.

Leo: I hate this. Ugh.

Steve: Or that have a crossing sign or something.

Leo: I hate helping Google's self-driving cars. Because that's what we're doing. You know that. For free. We're giving them free human input.

Steve: Funny you should say that because we may be helping Google with their code in this case, debug their code. This CAPTCHA says: "Select all squares with bugs. If there are none, click Skip."

Leo: This is a joke now. Come on.

Steve: It's like, well, I don't know, Leo, that second one, that second line down, third over...

Leo: There's a lot of hex in coding here. I don't know.

Steve: We've got unbalanced parentheses, I think, in that "if" clause.

Leo: Oh, my god.

Steve: So, yeah, anyway, I thought - someone sent me this, and I thought, oh, that's kind of...

Leo: That's hysterical.

Steve: That's fun. That's definitely way out there in upper geek land gets the humor of that.

Okay. So this came from Elaine yesterday when she sent back the transcript. "Hi Steve. I've been using MS Security Essentials since it came out, through XP and 7. Wikipedia says: 'Although support for Windows 7 ended on January 14, 2020 and MSE is no longer available to download, Microsoft will continue to update virus definitions for existing users until 2023.'" And so then she said: "Guess I'm good for a while. I still get new definitions every night." And I can confirm that I, too, am still getting nightly updates on my Windows 7 machine, and my MSE is continuing to scan and protect my Win7 machine.

So all of that, that we've been saying and grumbling about the last few weeks about, oh, the end of AV, and what are we going to switch to, and should we just go with nothing, just go commando. Turns out, no. Yes, our Windows 7 machines no longer get patches. But Microsoft is going to continue keeping the virus scanner, Windows Security Essentials or presumably Defender, depending on what you have in your Win7, current. So yay for that. And I just wanted to correct the record, that this is not going to be a problem. We all get to keep our virus things updated.

And, frankly, that's the biggest issue. Most of us are using non-Microsoft browsers, so Microsoft may choose not to update their browser. That's fine. Google's going to keep Chrome updated, they've said. Mozilla will be keeping Firefox updated. We have the commitment from them for that. And if we also have our in-built native AV still getting definitions for another three years, we're good. I mean, yes, if there's some horrible problem with Internet connectivity, I mean, you know, who knows what. Of course we'll

be keeping our eye on that and see whether anything that happens in Windows 10 could affect 7. But for what it's worth, we don't need to go searching for some other AV tool. So thank you, Elaine, for the news.

We did get a fix from Microsoft for the problem we discussed last week on Friday. That was the 7th of February. They dropped an out-of-cycle update to fix the desktop wallpaper stretch black screen of death problem which got introduced in the January Patch Tuesday. There's really nothing to their update. What's interesting is I was curious to know whether they were going to drop a formal Patch Tuesday on Windows 7 customers, but so far I checked this morning, and there were no updates for my Windows 7 machine. I meant to check again just before the podcast, but I got...

Leo: Have you had the problem that other people are reporting now where you can't shut it down?

Steve: Aha, that's what we're coming to.

Leo: Okay.

Steve: We're going to get to that in a second.

Leo: Okay, yes, because they've got more to fix.

Steve: They sure do. And I'll be surprised if that one doesn't get itself pushed out. So they said, of the wallpaper stretching problem, they said: "This update resolves the following issue." So just so people know, you've got to go get it. It is - and I don't have the number. I must have it here somewhere. Oh, yeah. KB4539602. So if you just want to be able in the future, even if you're not stretching a bitmap today, you may want to do that some day. So KB4539602 will allow you to do that, fixes that problem.

And they said, "This update resolves the following issue: Addresses an issue that might cause your wallpaper that is set to Stretch to display as black." Then they said: "Important." And, boy, is it. "Before you apply this update, see the prerequisite section." Well, now, it's not such a - I don't think it's that big an update in this instance. But we'll get there. So the prerequisites are that: "You must have the following updates installed before you apply this update."

And there are two. There's the one that we talked about way back earlier last year, the SHA-2 update. Remember that until I think it was June or July of last summer, the updates were double-signed. All the updates from Microsoft carried both an SHA-1 and an SHA-2 signature. And of course we were just talking about how SHA-1 has finally pretty much collapsed under continual academic pressure.

So Microsoft realized, well, it's not good to co-sign an update where one of the co-signatures is now known to be weak. So anticipating that, they removed SHA-1 signatures from their updates. The problem was that Windows 7 needed to be informed about SHA-2 updates. Unless you received this particular knowledge base update, which is 4474419, you would not be - your Windows 7 would reject the updates as being invalid because they weren't signed with SHA-1, and it hadn't been taught yet about SHA-2. So that's the first of the two.

The second one is kind of mysterious. It's from March of last year, and all they're calling it is a Servicing Stack Update, an SSU, Servicing Stack Update. And that has to be in place also. And I should mention that when I did, over the holidays, at the beginning of the year, I did a big update catch-up for GRC's servers. And I'm slow in installing updates for reasons we're about to get to here in a minute, relative to server. And so I'm kind of glad. And of course I also have lots of other rings of security surrounding GRC's servers. So they're protected in ways that depend much less on Microsoft than a server that's just sitting there with remote desktop protocol exposed to the Internet, for example.

But the point was my ability to get current was crippled by a mystery. I was unable to update one of my two servers. And it was finally, when I went back in my logs and saw that this Servicing Stack Update had not installed, that I thought, oh, well, okay. The Internet is telling me I need that. So I manually installed it, and then I was able to bring my machine current. So what's interesting is that Microsoft is saying that you will have automatically received this in Windows Update. But at the same time, they're saying, but if not, make sure you do. Well, I don't know what the story is, but I didn't get it automatically, and apparently people don't. So you need to have those both in order to fix the wallpaper stretch problem.

However, it turns out that after installing the fix, the 4539602 update, and I was supposed to know what that one was, that was not - oh, yeah. The wallpaper, the black wallpaper stretching fix. So, okay, first of all I should just say, I don't know who is going to have a stretched bitmap on a server. I guess people just think of it as Windows. I just sort of think of my servers as special. I don't, like, install all kinds of junk on them. And I don't care what the desktop looks like. I'm rarely seeing it.

But anyway, people applied the fix on Friday and then began getting a notice that you referred to, Leo. Oh, wait, no. You referred to the "can't shut it down." This is way worse. After installing the fix for - and this is just, like, beyond comprehension. After installing the fix for your bitmap stretch being black, that renders Windows Server 2008 R2, which is, as we know, the server version of Windows 7, unbootable. It will no longer boot.

Leo: But at least your wallpaper's not black.

Steve: Yeah, exactly. Now you're no longer being bothered by a black desktop.

Leo: You've got to really wonder. Good lord.

Steve: Unbelievable. On any instance of Windows Server 2008 R2 which is lacking those prerequisite updates I noted above, the consequence of attempting to install 4539602 to fix your wallpaper isn't a notice of an update failure or some nice mention about prerequisite updates missing. No, the result is a fully bricked server. For reasons only Microsoft knows, attempting to fix the desktop wallpaper stretching issue results in the deletion of two critical boot files that live in...

Leo: Oh, my god. This is so awful.

Steve: It's unbelievable. Winload.efi, think that might be important, and winload.exe are deleted from your Windows system.

Leo: This should in no way be coupled to anything with wallpaper.

Steve: No. No. It's like, what? So anyway, so as a consequence, since Friday, people were installing 602, and their system wouldn't restart. So the community has come up with some solutions. If you boot into the system recovery mode, then that'll kind of get your system basically going. You can get to a command prompt. If you have other Windows 2008 R2 servers that have not been bricked, they'll still have those two files. So if you can copy those files back into the Windows system32 directory, you're recovered. That's all you need to do. Or you can use the Windows imaging command line, the system imaging command "dism." And I have the command in the show notes here if anyone wants to take that approach and has these problems and hasn't already figured out how to fix it: dism.exe /image:C:\ /cleanup-image / revertpendingactions. And so that's sort of a manual way of undoing what it was that Windows inadvertently did that will bring your system current again.

And again, in my own experience, I mean, I could have easily fallen into this, too, because one of those prerequisites, that Servicing Stack Update, was missing. And it wasn't being given to me. I mean, I had other updates subsequent to March when that thing was released, but it just kind of got forgotten. And in this case that's not good. And the hits keep coming because, Leo, as you said, another problem has been afflicting people since the "final," maybe not so final Patch Tuesday update for Windows 7, which broke the wallpaper. Some people - and you've got to love the irony of this one, since Microsoft has been frantically working to push everyone off of Windows 7 and over to Windows 10. But now some Windows 10 users are being told: "You don't have permission to shut down this computer." And I have a picture of...

Leo: You bad man, you.

Steve: Oh, my lord.

Leo: That's crazy.

Steve: That is just unbelievable.

Leo: It really underscores how it must be the most horribly written program of all. I mean, ridiculous.

Steve: Yes. I had a friend, a long since ex-coworker who was at Microsoft. He was at Berkeley with me. Super smart guy. Oh, no, wait. No, I think [Loren] was definitely an MIT person. So he went to MIT. Anyway, we were talking about the state of Windows many years ago. And he just said, oh, it's just oh, oh.

Leo: If you only knew, Steve. If you only knew. Had he seen it? Had he seen the source code? Did he work at Microsoft?

Steve: Oh, yeah, he was deep, deep in. And I had another friend I mentioned who was the author, the originator of the whole .NET concept, the idea of a common language runtime was the brainchild of - actually he was an ex-employee of mine. And similarly,

just like, oh, we're just trying to leave that behind as quickly as we can. And think about it. I didn't have the update for today, that is, ready for today's podcast. But I did notice 99 fixes in today's Patch Tuesday. This is the second Tuesday in February for all Windows 8 and Windows 10 users, 99 things fixed including an IE zero-day, another problem in Internet Explorer actively being exploited in the wild.

And as I said, grumbling, last week, about how they're constantly changing it? Well, we know, if they won't just leave it alone, get their hands off it, it is never going to get fixed. They're going to just - this is life now is this moving, as I called it, a "smear," a Windows 10 version smear, because they're just constantly changing it.

So anyway, a number of workarounds have been found for people who are being told they no longer have permission to turn off their own computer. If you log off, that'll bring you back to the logon screen. And as we know, down in the lower right there is the option to shut the computer off. You can do that. Apparently you can do CTRL-ALT-DELETE a few times to get to a similar screen where powering off is an option. That'll avoid this. And there is a group policy edit tweak that you can apply, basically giving yourself permission - imagine that - to turn off the computer.

Again, are we going to see an out-of-cycle update? Is Microsoft going to say, okay, maybe we ought to just push an update out to people? I don't know. I don't know what their policy is.

Leo: It's almost guaranteed to break something.

Steve: Maybe they unplugged all of those Windows 7 update servers, and they're just gone now. Or they said, let's send them to the Azure cloud. Send them off to a better place. I don't know.

Leo: Unbelievable.

Steve: Wow. And then just, you know, because this just bugs me, Windows 10 Firefox users are being reminded about Edge. So I figured, while I'm on the topic of Microsoft and Windows 10, I suppose anyone who hasn't deliberately turned off the "suggestions" option, I mean, that's like - it's not Candy Crush Soda Saga, but it's up there. I mean, you know, when presented with a switch from Microsoft about would I like to have some suggestions, like, no. You have nothing to suggest that I want to know about. So of course mine's off.

Anyway, so this is on the Start menu. And maybe you didn't know you could turn it off, or maybe you left it on because you are interested in what Microsoft may have to suggest. But, you know, as we know, Windows 10 is now free and intended to be a source of, in Microsoft words, "significant marketing and profit opportunities moving forward." So this is what we get with this new approach towards an operating system.

And I'll just say it's not for the sake of running Windows that we run Windows, exciting and harrowing thought it can sometimes be. Windows is a means to an end. It exists to host and launch other programs. It's an operating system. So it seems a bit unseemly for people with their Start menu suggestions still enabled to receive the following selective notice when Microsoft is their deliberately chosen browser. What Firefox users are now getting on their Start menu...

Leo: Oh, this pisses me off.

Steve: Yes.

Leo: I'm sorry. Did I say that out loud?

Steve: Yes, you did. Thank you, Leo, I appreciate your report.

Leo: It irritates me no end.

Steve: Isn't that wrong? Doesn't that feel wrong? So up at the top of your Start menu, under "Suggested," it says, with the new surf wave Edge logo, "Still using Firefox?"

Leo: What's wrong with you?

Steve: Microsoft Edge is here.

Leo: Oh. Ugh.

Steve: You know?

Leo: Now, does it say it just once?

Steve: Well, let's hope. I don't know. Once is enough. That just...

Leo: Yeah.

Steve: Okay, so, you know, every week these notes that I publish, that you're looking at, that I'm reading from, that our listeners can download, they're authored in Google Docs on Chrome because that's the best way I've found of doing something like this. That's alongside a Firefox browser, with a long vertical column of tabs, where all of the news of the week I've pulled together and sorted and arranged and put them in, you know, so like pulled what I want to talk about this week together. And there's an instance of the ThoughtManager Desktop outliner app.

You know, I'm working toward a tentative peace with Windows 10 because Microsoft really hasn't left me or anyone else, any of us, with any practical choice unless we want to just leave the Windows universe. But I don't need their help choosing the best tool for the job. I'm delighted Edge has incorporated Chromium. But to answer your question, yes, Microsoft, I'm still using Firefox.

Leo: Yeah. Yes, I am. You got a problem with that, buddy?

Steve: I know. Oh, boy. That just, you know, well, it's part and parcel of Candy Crush Soda Saga. It's not there because they want it there, it's there because they're getting paid. And in this case they're just annoyed that, I mean, I guess I could see the logic. Maybe somebody used Firefox because the previous Edge didn't work for them somehow. So they're saying, oh, like, we fixed it. Now it actually is a good browser. I don't know.

Leo: Yeah. I don't know. I don't know. You know, they're, well, we talked about this before. They're making some enterprise users change their search from Google to Bing.

Steve: Yes, yes.

Leo: I mean, this is - there's somebody at Microsoft doesn't really understand user freedom or something. I don't know. I don't get it.

Steve: Well, I mean, it is the direction they're going in. And, you know...

Leo: I don't know. I wouldn't say unilaterally. The company also supports open source and, I mean, I don't know.

Steve: And they do have Edge based on Chromium, which is, you know, a good thing.

Leo: Right. There's somebody in marketing that's just annoying. I don't know. I don't get it.

Steve: So last week Google closed an Android remote code execution flaw which was discovered in the Bluetooth daemon running in Android. It has been patched. It was patched last week in Android's February security update. And, you know, we've been encountering Bluetooth flaws recently. And while they're not good because they are potentially hands-off and at a distance, at least the deliberately lower power short-range operation of Bluetooth tends - and of course we know that's not an absolute, either, because you can get directional antennas and Pringles cans and things - the short-rangeness tends to limit the vulnerability's exportability and severity. Certainly WiFi vulnerabilities are worse. And Internet TCP flaws are worse still because they work on a global scale.

But in this case a critical vulnerability was found and fixed in the Bluetooth implementation on Android devices which could allow, which is to say does allow, attackers to launch remote code execution attacks without any user interaction. So it's one of those bad ones where the user who had been compromised would not know it.

Last Thursday, after the patch had been pushed out, the researchers who found it revealed additional, but not all, because they're trying to be responsible, details behind this flaw. It's tracked as CVE-2020-0022. It poses a potential critical severity threat to Android versions Pie, so that's 9, and Oreo, 8.0 and 8.1, which account currently for almost two thirds of Android devices today, assuming that they've got Bluetooth enabled, as most Android devices probably do for the various, I mean, I even find it turned on on my things when I've explicitly turned it off.

We've talked before about how Apple seems to think they know better, and they keep turning Bluetooth back on for me. It's like, okay, well, I don't need that power drain, and I don't have any Bluetooth things hooked up, and I'd rather not have the risk of one additional radio thing which could have a problem, exactly as this does. And that's been our advice on this podcast for years. If you're not using a radio, whatever it is, turn it off because it's not helping you. It's consuming some power, and it creates an inherent vulnerability at a distance.

Leo: It's analogous to don't turn any services on on a computer unless you know you're going to use them.

Steve: Exactly.

Leo: You're just opening it up for - yup.

Steve: So against 9.0 Pie and Oreo, Pie and Oreo 8.0, 8.1, and 9.0. The researcher said that a remote attacker within Bluetooth range can silently execute arbitrary code with the privilege of the Bluetooth Daemon, and it runs in the kernel. The flaw is worrisome because no additional interaction is required, and only the Bluetooth MAC address of the target device needs to be known to launch an attack.

Okay. So, well, there are a couple reasons that's not comforting, because it turns out that for many devices the Bluetooth MAC address can be deduced from the WiFi MAC address. They're often sequential. And so WiFi is easily known. It's being broadcast by the smartphone's WiFi. So obtaining the Bluetooth MAC address is probably a matter of adding or subtracting one, depending upon which phone you're using, and maybe they're all the same. I haven't looked.

The same vulnerability does impact Google's most recent Android v10. However, with Android 10, the severity rating is dropped to moderate rather than critical because the impact is not a remote code execution as a consequence of other changes made in Android 10. It will crash the Bluetooth daemon, but it won't give you remote code execution access. And they did not test any Android versions older than 8. So we don't know either way whether those may be affected. The flaw's discoverers said they are confident all patches - they said, sorry, once they are "confident" - and I put "confident" in quotes in the show notes because you'll see where I'm going - all patches have reached the end users, they will publish a technical report on the flaw that includes a description of the exploit as well as proof of concept code.

The trouble is all of us here know that a great many Android devices running Oreo and Pie are never going to receive an update. So, I mean, even Windows systems that have automatic updates universally applied somehow manage not to receive them. So we know that many suppliers of lower Android devices aren't being responsible with pushing updates out to their customers. So those people who are not receiving updates for Oreo and Pie, two thirds of the current Android install base, we don't know what percentage are receiving updates, they will now probably forever be vulnerable to the possibility of an engineered proximity takeover and malware installation. And we know that completely descriptive documentation including a working proof of concept, will be made available shortly. Maybe they'll wait a week. Maybe they'll wait two weeks. But maybe a month, doesn't matter, a huge percentage of devices are not going to get fixed.

And this is precisely the sort of powerful and persistent vulnerability that the powers that be, hostile powers, border shenanigans, crossing into China, wherever, where people

say, "Oh, yeah, I got some stuff installed on my phone." Well, this is going to - they're like, here's a new way for that to happen for anybody whose phone is sufficiently vulnerable. They didn't even have to take your phone behind a screen somewhere. They could just, you know, figure out that this is you, or maybe just try everybody who's moving through and see how many phones they're able to install some backdoor spyware onto.

So this is really bad. We've said it before, and it bears repeating. Today's smartphones are seen by bad guys as a huge target of opportunity. And just as no one today wants to use an operating system that's no longer receiving security updates, people should be reluctant in the extreme to use any smartphone whose manufacturer does not have a solid track record of providing updates. It's true that such after-sale support comes at a cost. The cheapest phones won't have it. But in this case, you really are getting something valuable for the money.

So I just can't imagine, Leo, using a phone that is not from a major manufacturer that is known to be responsibly putting out updates. And to be doing it for the service life of the phone. It's not good enough to say, oh, we're going to do it for three years, but you keep using the phone afterwards. Or as an end user, you have to have the self-control to say, okay, updates have stopped. I'm going to update to a newer phone. You just have to.

Jonathan Knudsen, who is the senior security strategist at Synopsis, said of CVE-2020-0022, he said: "It can be exploited by anyone within range of your vulnerable phone who can determine your Bluetooth MAC address, which is not difficult." He said: "As a user, keeping current with updates and applying them in a timely manner is important. Unfortunately, many vulnerable, slightly older phones will not have continuing software update support from the manufacturer, which means users are faced with two unattractive options: either disable Bluetooth entirely" - and certainly that's a good way to do it, and make sure it stays off - "or get a newer phone."

So the February patch roundup for Android included patches for 25 bugs, with 19 of those vulnerabilities rated high. There were four others that were high, but they were specifically tied to Qualcomm chipsets used inside Android devices. So this was the most worrisome of those. If you happen, if you're listening to this, and you've got an Android phone that didn't get updated last week, you need it to be updated or turn Bluetooth off, if you're concerned that, you know, if you're a target of opportunity, if you're a little unnerved by the idea that in a couple of weeks full disclosure will be provided; all the bad guys in the world will know how to do this to any Android devices. Not that old, either. Pie and Oreo that haven't had updates. It's just you can't use a smartphone that isn't on the update flow.

So last week we talked about the Clearview AI company, who were doing the facial recognition and bragging that they scraped the web for three billion faceprints and made them available to 600 police departments so they could identify people within seconds. Since then, Clearview has increased their collection of cease-and-desist letters, which was not exactly what they were hoping to be collecting, from major U.S. social media players. The first one they received was from Twitter a couple weeks ago, when Twitter told Clearview to stop collecting its data and to delete whatever it had. In addition, Facebook has similarly demanded that Clearview stop scraping photos because that action violates Facebook's policies. And now Google and YouTube are also both telling Clearview to stop violating their policies against data scraping.

Clearview's take on this is defiance. The CEO, Hoan Ton-That, was interviewed last Wednesday morning on CBS's "This Morning" news show. He told listeners to trust him. He said the technology is only to be used by law enforcement, and only to identify potential criminals. Ton-That claims that the results, which is not encouraging, are 99.6% accurate. I guess, though, you wouldn't want a false positive to misidentify you as

a bad guy. So I guess accuracy is a better thing. And he also claimed that it's his right to collect public photos to feed into his facial recognition archive. He said: "There's also a First Amendment right to public information. So the way we have built our system is to only take publicly available information and index it that way."

Leo: And, by the way, there was a recent Supreme Court decision having to do - was it the Supreme Court? Maybe Ninth Circuit Court - having to do with scraping of LinkedIn in which they ruled, yup, you can't stop scraping. If it's public information, you can't stop it.

Steve: In fact, I have that, I have a mention of that here. So we know from last week when we talked about this that in Illinois, at least, with their BIPA, the Biometric Information Privacy Act, it's illegal there. And YouTube's statement read: "YouTube's Terms of Service explicitly forbid collecting data that could be used to identify a person. Clearview has publicly admitted to doing exactly that. And in response, we sent them a cease-and-desist letter."

As for Facebook, Facebook said last Tuesday that it has demanded that Clearview stop scraping photos because the action violates its policies. Facebook said: "We have serious concerns with Clearview's practices, which is why we've requested information as part of our ongoing review. How they respond will determine the next steps we take." Which I'm sure Facebook intended to sort of sound ominous. And Ton-That defended Clearview as being a Google-like search engine. He said: "Google can pull in information from all different websites. If it's public, and it can be inside Google's search engine, it can be in ours, as well."

Google disagreed, saying that Clearview isn't at all like their search engine. Google said: "There's a big difference between what we do and the way you're shanghaiing everyone's face images without their consent. Most websites want to be included in Google search, and we give webmasters control over what information from their site is included in our search results, including the option to opt out entirely." Google said: "Clearview secretly collected image data of individuals without their consent, and in violation of rules explicitly forbidding them from doing so." So the question is, when is public information not public?

Which brings me to the point you raised, Leo. Back in 2016 a company called hiQ, which I recall we talked about at the time, a San Francisco startup was marketing two products which depended upon whatever data LinkedIn's 500 million members had chosen to make public. There was the first product they called "Keeper," which identified employees who might be ripe for being recruited, and "Skills Mapper" summarized a LinkedIn member's skills. In that instance, hiQ was amassing public information, grabbing the same material that anyone could get from Linked In, without having to log in. So any browser would display the same information hiQ was vacuuming up, organizing, and reselling. And when sufficiently analyzed, inferences could be made to alert companies, for example, when their pivotal employees might be interviewing for another position. And you can do much more, as we know, with this kind of advanced informatics.

Leo: Oh, geez.

Steve: Yeah, isn't that interesting. You put a flag to be notified when any of your employees' LinkedIn profiles indicate maybe they're, you know, that longer lunch break was a little more than a lunch break.

Okay. So in the case of hiQ, LinkedIn sent a cease-and-desist letter alleging that it was violating serious anti-hacking and anti-copyright violation laws. And LinkedIn cited the Computer Fraud and Abuse Act, the CFAA; the Digital Millennium Copyright Act, the DMCA that we've talked about so much; and California Penal Code section 502(c), whatever that is. LinkedIn, and this is a little interesting aside, had been exploring how to do the same thing with its own data that hiQ had achieved, also noted that it had blocked hiQ from accessing its data. And, as you mentioned, it was just this past September, in 2019, an appeals court told LinkedIn to back off, and that it had no legal right to interfere with hiQ's profiting from its users' publicly available data. The court protected data scraping of public data in what at first looks like a major legal precedent, but it's actually a lot less clear.

Our friends at the Electronic Frontier Foundation wrote: "While this decision represents an important step to putting limits on using the CFAA" - the Computer Fraud and Abuse Act, because the concern has been that it's being abused - "to intimidate researchers with the legalese of cease-and-desist letters, the Ninth Circuit sadly left the door open to other claims, such as trespass to chattels or even copyright infringement, that might allow actors like LinkedIn to limit competition with its products." So essentially the Ninth Circuit didn't go as far as our EFF folks wished it had. But in this case at least it said, hey, you, LinkedIn, are not lawfully allowed to prevent somebody from visiting your site with automated scrapers and obtain whatever information has been made public by your users.

And of course the problem is the CFAA, the Computer Fraud and Abuse Act, is broadly written and subject to multiple conflicting interpretations across different federal circuits. This makes it likely that the Supreme Court will ultimately be forced to resolve the meaning of terms which are not really clear. The CFAA says, for example, "without authorization." Well, people want to take without authorization the way that they want to. And this of course is bad. This occurs with broadly written legislation that just ends up having to - it's a problem for everyone and ends up having to get resolved in the courts.

The EFF's Surveillance Litigation Director - there is such a person. Their Surveillance Litigation Director Jennifer Lynch said that Clearview is the latest example - so now we're talking about Clearview, the facial scraping people. That doesn't sound right, facial - well, anyway.

Leo: Yes, it's a laser procedure.

Steve: Yes, you can get that done.

Leo: Makes your skin soft.

Steve: Yeah, it's covered by insurance - is the latest example of why we need laws that ban or at least pause, pending more clarification, law enforcement's secret abuse of facial biometric recognition. She cited many cases of what she called law enforcement's, and Clearview's specifically, abuse of facial recognition, stating: "Police abuse of facial recognition technology is not theoretical. It's happening today. Law enforcement has already used live face recognition on public streets and at political protests."

And of course as we've observed before, this is all being enabled by the recent incredible reductions in cost. The cost of processing power has crashed. The cost of mass storage has collapsed. The cost and presence of ubiquitous networking communications, it just doesn't cost anything anymore to send data, massive data all over the place. We didn't

have this 10 years ago. And Leo, although the podcast will have run out by 10 years from now, we'll still be around, and we'll see what we have 10 years from now.

Leo: We'll see if anybody recognizes us.

Steve: The NIST is testing methods of recovering data from smashed smartphones. And, you know, it makes sense when you think about it. There's been a lot of discussion through the years about how to best irreversibly kill a hard drive. And we talked recently, not too long ago, about this. One of my favorites, since many modern hard drive platters are now being made of glass, you can often take a hard drive, kind of like a - what was it we used to smash? Bit-O-Honey? No, no, not Bit-O-Honey.

Leo: With a little silver hammer?

Steve: Yes. You could take a...

Leo: That was Bit-O-Honey.

Steve: There was that big white thing that was chewy.

Leo: Laffy Taffy?

Steve: I remember, yeah, you'd smash it.

Leo: Bonomo. It was Bonomo. And, yeah, you'd smash it, and it'd be little pieces, and you'd eat it.

Steve: Yes. And then you peeled the paper off. And it's like all got little itty-bitty chewy...

Leo: Yeah. It was good. Loved that.

Steve: Yeah. Anyway, turns out you can do that with many hard drives. You take a hard drive, just smash it face down onto the concrete or asphalt or something, a hard surface. And then, if you shake it, put it up next to your ear and shake it, if you hear the sound of lots of little bitty fragments, then you know...

Leo: How do you know you got all the platters?

Steve: That's true. Chances are...

Leo: I know this is true because 20 years ago Patrick Norton, we were showing people how to destroy hard drives, and he opened up a hard drive. See, because usually it's pretty easy to unscrew a hard drive.

Steve: You just need a torque spring, yeah.

Leo: Yeah. Pull out the thing. There's three, four, five platters. In the case of the 16TBs, I don't know, 28 platters. And then he said, and watch, I can just - and he hit it with a hammer. He didn't know it was glass. He thought it was going to be metal. He was going to bend it. And shards of glass flew everywhere. We weren't wearing protective eyewear. This is live television. We're very lucky. Talk about face scraping, I mean, it was - anyway, yes. So this is nothing new. They've been made out of glass, in some cases anyway, for a while.

Steve: Yeah. It just turns out glass is a fluid, and it is an easier substance to work with to get the level of smoothness that you need...

Leo: Oh, yeah, of course, yeah.

Steve: ...in order to fly heads as close to these things as they are. But what about an entirely solid-state smartphone? We talked about it just a couple weeks ago. One of those guys had shot one of his two smartphones, and the FBI claimed to have brought it back to life.

Leo: Yeah.

Steve: Which I think is a miracle.

Leo: By the way, you got any Bit-O-Honey? Because I've got a hammer.

Steve: The bad guys, it turns out, this is a thing. Bad guys are now smashing their phones, drowning them in water, shooting them with a gun.

Leo: What did they do - doesn't he throw it in the microwave on "Mr. Robot"? Oh, those were the little SIM cards he'd throw in the microwave.

Steve: Yeah. You don't want to do that. That would be rough. You used to be able to put a CD, an audio CD in the microwave.

Leo: Yeah, it would spark.

Steve: And it would make all kind of sparkliness and things, yeah.

Leo: Don't do that at home, kids. It's not good.

Steve: So the question is, how effective is physically destroying a smartphone? It turns out that many criminals have discovered to their chagrin that reducing their devices to smashed plastic and glass means nothing, if the device's little black epoxy memory chips have managed to survive. Forensic engineers who work with police to gather evidence have become quite adept at performing, like, amazing feats of posthumous data extraction. With more and more evidence now sitting on smartphones, a better understanding of what works and what doesn't has been turned into a growing issue of some urgency.

So our U.S. National Institute of Standards and Technology (NIST) recently conducted a series of tests using 10 popular Android smartphones which had accumulated a mix of data during their simulated use. The NIST engineers and their forensic partners then attempted to extract the data from the surviving memory chips using various methods to compare with the original dataset. In some cases the chips could be left attached to the original motherboard and accessed via the JTAG serial interface, which all systems have. JTAG is an industry standard serial communications protocol for testing and programming electronics. In other cases, the chips were physically removed from their original motherboards and then interconnected to directly, sort of an in vitro data extraction.

So the NIST wrote in their report: "The comparison showed that both JTAG and chip-off extracted" - that's what they call it where they have to remove the chip from the board - "were able to extract the data without altering it, but that some of the software tools were better at interpreting the data than others, especially for data from social media apps."

And as I was reading this and thinking about it, I thought, that's a good point. It's one thing to have access to the raw - presumably unencrypted, I think that's why they chose Android phones - data. But you still need to be able to make heads or tails of what you have. I mean, it's a chip, you know, so it's like, okay, here's the contents of this grid of bits. Now you've got to make sense of it. Either way, it's a big ask, to tell some guy, okay, here's a destroyed phone. We need to know what data is in here.

It turns out that there are trained forensics people whose days are spent doing this. They have an expert at NIST, Rick Ayers, who said: "Many labs have an overwhelming workload, and some of these tools are very expensive. To be able to look at a report and say this tool will work better than another for a particular case can be a big advantage." So essentially they're sort of trying to create some decision framework for forensic data recovery.

What really piqued my interest was that Cellebrite, the company and the technology that we've spoken of often here, was one of the two systems that was used. I've got a link to a PDF in the show notes. The PDF is titled "Test Results for Binary Image JTAG and Chip-Off Decoding and Analysis Tool: Cellebrite Universal Forensic Extraction Device (UFED)." And that's an acronym that we've seen before and talked about. So they call it the Cellebrite Universal Forensic Extraction Device Physical Analyzer, and it's now at v7.20.0.123. And it is interesting to scroll through this. They're located in Parsippany, New Jersey, at 7 Campus Drive. We know that from the report.

And the results summary said: "Cellebrite's Physical Analyzer is a versatile mobile forensic solution that runs on existing hardware. It comes with a suite of applications, peripherals, and accessories. Physical Analyzer was tested for its ability to decode and analyze binary images created by performing Chip-Off and JTAG data extractions from supported mobile devices. Except for the following anomalies, the tool was able to

decode and report all supported data objects completely and accurately for all mobile devices tested."

And so what we have is a list of a few exceptions for - there were some standalone files for an HTC One Mini in the chip-off that I guess it had a problem with. There were some social media-related data that an LG K7 chip-off extraction had a problem with, the ZTE 970 chip-off had a problem with, and that related to some Twitter data that they could not recover. The HTC One XL where the chip had been removed, the HTC Desire S where its chip had been removed, and then two HTC phones where JTAG, the JTAG serial interface was used had some problem reconstructing some Facebook data.

But by and large, we're talking about a contact, well, it says in the report here they were able to recover and perfectly reconstruct deleted contacts, calendar, memo note entries recovered from the HTC Desire 626, ZTE 970, the Moto-E, the Samsung S2, the HTC One XL, and the Samsung S4. They were able to pull deleted contacts and calendar entries from the LG K7 and HTC Desire S. Deleted contacts and memo entries were recovered from the HTC One Mini. Deleted call logs were recovered from the LG K7, the Moto-E, Samsung S2, Samsung S4, and HTC Desire S. They were able to pull deleted SMS entries recovered from the HTC Desire 626 and a bunch of others, and bookmark entries recovered from the HTC Desire 626 and others.

So I thought this would be interesting to our listeners because, I mean, this demonstrates that you can't crack your phone in half or apparently even shoot it with a bullet. You need to, if you were someone, I mean, even for benign purposes, you know, these are reconstructed deleted data from the memory is being completely recovered by this Cellebrite forensic analysis tool. So this stuff is real. And essentially what it means is you need to take your phone apart and get your drill and drill a hole through all of the little black chips that you see on your phone, if you really and truly want to keep solid-state memory from being recoverable. These are not forensically wiped. If you were able to do a really good forensic overwrite of the data, then that would have rendered them unrecoverable.

But failing that, you really need to reduce solid-state storage to a state where the individual components of it are clearly destroyed. Otherwise, if anybody had sufficient motivation to reconstruct the data, apparently this is something that is now just, I mean, there are people who spend their days doing this. And the NIST has, oh, the other issue that I didn't bring up that was mentioned was the issue of the chain of evidence. So in order for a defense attorney not to be able to poke holes in this, it's necessary for this to be done in conditions where the chain of evidence is not broken. So labs need to be certified, and the phones need to be handled appropriately and so forth.

But the point is it is really and truly a thing to be able to recover data from a phone, even if it really looks very sad. It may still have some vital components that are intact, and that's all it takes. Which, wow, you know, it's just like it really does happen.

Okay. And on the brighter side, actually, that's a pun because the paper was titled "Brightness." This is from our Yet Another Data Exfiltration Technique of the Week department. The title of the paper: "Brightness: Leaking Sensitive Data from Air-Gapped Workstations via [yes] Screen Brightness."

Leo: Oh, my god.

Steve: I know. But these are the guys at the Ben-Gurion University and the Department of Electrical and Electronics Engineering at the Shamoon College of Engineering, both in Israel. These are the guys that have done some amazing stuff before. We'll talk about

that in a second. Their abstract from their paper, I have a link to the PDF here in the show notes, they said: "Air-gapped computers are systems that are kept isolated from the Internet since they store or process sensitive information. In this paper we introduce an optical covert channel in which an attacker can leak, or exfiltrate, sensitive information from air-gapped computers through manipulations of the screen brightness."

Leo: It makes sense, though. It's like a semaphore. Yeah.

Steve: Yeah, yeah. "This covert channel is invisible, and it works even while the user is working on the computer. Malware on a compromised computer can obtain sensitive data - files, images, encryption keys, passwords, whatever - and modulate it within the screen brightness, invisible to users. The small changes in the brightness are invisible to humans, but can be recovered from video streams taken by cameras such as a local security camera, a smartphone camera, or a webcam. We present related work and discuss the technical and scientific background of this covert channel. We examined the channel's boundaries under various parameters, with different types of computer and TV screens, and at several distances. We also tested different types of camera receivers to demonstrate the covert channel. Lastly, we present relevant countermeasures to this type of attack."

Okay. So first of all, what's fun about these guys is they take strange things and, like, really wrestle them all the way to the ground. We've talked about the topic before. And this appears to be a particular hobby horse for these guys. In previous years we've covered their serious research into all manner of air-gapped computer data exfiltration. They're the guys, they talked about ways of getting data out through PC speakers, blinking LEDs, infrared lights in surveillance cameras, and - remember this one, Leo? - even modulating the rotation rate of a computer's cooling fans.

That was their famous, they called it "Fansmitter" research. And it demonstrates that, yes, where there's a will, there's a way. They were actually changing, slightly changing the fan speed and using the fact that that could be audibly detected as bits. And I don't remember now how many different levels of speed, maybe it was three bits, so they were like eight different speed levels. And so they were able to pull eight bits at a time. It wasn't fast, but they were able to do it.

So my first reaction to this was to wonder what computer containing data worthy of exfiltration through such measures would tolerate having its screen within the view of a camera of any kind. So that seemed a little bit farfetched. You needed to have an environment where that could happen. But they have a seven-page research paper, and they tackled the problem with their usual thoroughness, just as they tackled the question of how many bits per second can we transmit with the sound of a fan's speed being varied.

Anyway, they conclude, undeterred after seven pages: "In this paper we present an optical covert channel in which data is concealed on the LCD screen brightness, invisibly to users." They talk about, I think it was a 3% change in brightness where that was enough to electronically detect it from a video recording of a surveillance camera while the user just sat there looking at it, but there was no obvious change to the user. They say: "We exploit the limitations of bare human vision concerning brightness perception, using sufficiently low values of contrast between the brightness levels. Consequently, the current results demonstrate the feasibility of our covert channel, while outlining its boundaries. Notably, this kind of covert channel is not monitored by existing data leakage prevention systems."

So, yup, you could slightly change the intensity of a screen in order to send a one or a zero. You'd need some fancy coding in order to do that. But self-clocking technologies for data exist. That's what hard drives use. So it's possible to do it, and these guys figured out what the maximum baud rate was. It would still take a long time. But remember that we do have many high-value secrets that are not very long, like an elliptic key. We like elliptic keys because they're short. They're much easier to handle. And they get processed more quickly. Unfortunately, they also get exfiltrated more quickly. So anyway, just another little wacky bit of data exfiltration research. So Leo...

Leo: Now let's talk about cookies, yes.

Steve: In a minute. I wanted you to tell our listeners about some news that just broke in the Washington Post.

Leo: I saw this.

Steve: I did not have a chance to come up to speed fully.

Leo: Big story, yeah.

Steve: But, yeah.

Leo: Okay. I'll let you go ahead, and then I'll get the story out.

Steve: Oh, okay. So, well, I guess I just wanted to mention...

Leo: Oh, you want to talk about it now.

Steve: Well, I don't - I didn't know how much you knew about it, if you knew anything more than just the headline that the CIA was behind basically a fraudulent crypto company.

Leo: We talked about it on MacBreak Weekly.

Steve: Right.

Leo: Since the '40s. Since the '40s, Crypto AG was, everybody thought, a Swiss company. But in fact it was owned outright by the CIA and West German Intelligence. They were making coding hardware, like Enigma machine-style things. And according to the Post, they rigged, the CIA and West German Intelligence rigged the company's devices so they could break the codes that countries used to send encrypted messages. The decades-long arrangement among the most closely guarded secrets of the Cold War is laid bare in a classified, comprehensive, CIA

history of the operation obtained by the Washington Post and ZDF, a German public broadcaster. It was codenamed "Thesaurus," later "Rubicon."

The CIA report concludes it was the intelligence coup of the century. Foreign governments were paying good money to the U.S. and West Germany for the privilege of having their most secret communications read by at least two and possibly as many as five or six foreign countries. The 1979 hostage crisis they were monitoring the Iran mullahs. They fed intelligence about Argentina's military to the British during the Falklands War. It's stunning.

But it underscores something I've said for a long time. The only kind of crypto you should use is open source. If it's a binary blob, if it's a black box, you don't know who's on the other side. You've got to use open source. And then at least Matthew Green can look at it and tell you, okay; right?

Steve: Yeah. Well, and of course that's the way we have to go with voting systems in this country.

Leo: Same thing.

Steve: It would be fine for Diebold to manufacture the hardware and to sell it. But they've got to be using something that has been heavily scrutinized. So it can still be a profit center. It just, you know, we just have to not have the software be part of what's proprietary.

Leo: Right. Exactly. Yeah, isn't that an amazing story? I thought you were going to - I thought you wanted to talk about the fact that Microsoft has apparently backed down on "Bing jacking," according to our good friends at Bleeping Computer.

Steve: Yay.

Leo: Yay. Microsoft - I knew they would do this at some point, backpedaling on forcing Bing search for Office 365 users. Mary Jo Foley and Paul will be doing a little dance. Microsoft says it heard customers' concerns by the way the company planned to roll this value out. So they're not going to do it.

Steve: Yeah, good. Good. And stop telling Firefox users to change browsers.

Leo: Yes. Criminently.

Steve: That's just wrong. Okay. So promiscuous cookies. This is relevant to us at the moment because Chrome 80 appeared last week with its implementation of the updated handling of an optional cookie property called SameSite. We first noted that this was happening last May. We talked about it briefly at the time. There was an IETF draft from Google which proposed a change to the default behavior for when cookie behavior was non-specified. This all revolves around third-party cookies. We've talked about them a lot. A third-party cookie is a cookie that the browser returns to a domain other than the one that provided the page that you're looking at.

So, famously, this is the way advertisers track us is that an advertiser presents their bit of content in a window on a web page. And even though it was never intended for this reason, browsers have always honored by default third-party cookies. Notably, Safari never has. I've always thought Apple was just on the ball for this. Of course it does demonstrate there are other ways to track people than cookies. But cookies is like the official means for maintaining state. But the ad is not the first party, which is the site that you're visiting. It's coming from a third-party server.

And then, of course, if you go to some other location, some other website that is also being served an ad from the same ad server, well, your browser returns the same cookie at this other site as it returned from the previous site. That links you. That's the way tracking happens. So the problem is there are other abuses, cross-site request forgeries, which are a real problem for web applications, which also involve sort of a different flavor of abuse of third-party cookies, not relative to tracking, but spoofing session state in a way that can be used to steal credentials.

So last May, in the abstract from a guy from Google, and this is what I remember sharing last summer, he says: "This document proposes two changes to cookies, inspired by the properties of the HTTP State Tokens mechanism proposed in" - and then there's another document reference. Now, we should mention that HTTP State Tokens is a "maybe we're going to someday get this" replacement for cookies. So I guess the point is that the engineers who are moving the web technology forward, they fully recognize that cookies, well, the world has changed dramatically since cookies were first created. Cookies have been overtaxed with the things that they're being asked to do.

So it would be wonderful if we could design a proper HTTP State Token mechanism to correctly do what it was that cookies were originally created by Netscape back in the beginning of all this to handle. Problem is change is difficult. I mean, even this change, the change we're talking about today, about cookie promiscuity, is turning out to be difficult, as we'll see. But so this guy is saying the document proposes two changes to cookies inspired by the properties of the HTTP State Tokens mechanism. What happened was, out of that discussion came an awareness that cookies could be fixed a little bit without throwing them out completely.

So he says: "First, cookies should be treated as" - and there's an expression - "SameSite=Lax by default." There is a, I can't think of the name, an attribute, that's the word I wanted, an attribute which cookies can be given. Cookies can, for example, be given the attribute of "secure," in which case no browser will send a cookie that it has for a site over a non-secure connection. In other words, HTTPS has to be present if the cookie has the attribute tag "secure."

Similarly, and there are a number of different types of attribute tags. For example, "expires" is another one, how long should the browser keep this before letting it go. And if there is no expires date or time, then it's automatically a session cookie, so that it will not save it in permanent memory. It will, as soon as you close the browser, the browser forgets that cookie, which can be useful in some instances. Another attribute is "SameSite." And that can have the values of, first of all, it could not be specified at all, or it can explicitly have the value of None, Lax, or Strict. And we should think of it as same-site enforcement. In other words, you could have no enforcement of same-site handling, you could have lax enforcement, or strict enforcement.

So this guy is saying that, first of all, he's proposing that the default, which has been None, that is, no same-site enforcement, should be elevated to Lax by default, which is a big change. That is, so if there's no specification, rather than the no specification meaning, okay, no change in same-site enforcement, then we're going to change this so that there will be a change. And he said, secondly, cookies that explicitly assert SameSite=None in order to enable cross-site delivery should also be marked as Secure.

So this guy is proposing these two changes. We're going to change the default to tighten the cross-site handling of cookies in two ways. And I'll explain the first one here in a second. The second of those is that, if somebody had explicitly said we want no change, SameSite=None, that from now on that will only be done over a secure connection. Even if secure isn't explicitly stated, that is, that becomes now part of the spec.

So I have a lot of detail here that it's difficult for me to - I've pretty much covered this already just verbally. If you have a cookie marked as Strict, then it will never be sent in a third-party context. And the way that could happen is you can receive a cookie in a first-party context. And then, if a query comes in, or if you then make a query to that domain for which you received a cookie in the first-party context, but that's a third-party domain, then if the cookie was marked as Strict for same-site enforcement, the browser pretends not to have a cookie, just doesn't return it to the server as part of the response.

So "The HTTP State Tokens proposal," this IETF document explains, "aims to replace cookies with a state management mechanism that has better security and privacy properties. That proposal is somewhat aspirational," he recognizes, says "it's going to take a long time to come to agreement on the exact contours of a cookie replacement, and an even longer time to actually do so."

He says: "While we're debating the details of a new state management primitive, it seems quite reasonable to reevaluate some aspects of the existing primitive," which is cookies. He says: "When we can find consensus on some aspect of HTTP State Tokens, we can apply those aspirations to cookies now, driving incremental improvements to state management in the status quo."

And so essentially what has happened is Google is the first browser with Chrome 80 to bite this bullet and make this change, to essentially change the way in some instances cross-site cookies are handled and returned. And I've scrolled through a whole bunch of stuff in the show notes that anyone who's really interested in the nitty-gritty can read.

Troy Hunt blogged about this pending change last month. On January 3rd he blogged a posting titled "Promiscuous Cookies and Their Impending Death via the SameSite Policy." The top of his blog, before he gets into a lot of details, he said: "Cookies like to get around. They have no scruples about where they go, save for some basic constraints relating to the origin from which they were set." He says: "I mean, have a think about it."

He said: "If a website sets a cookie, then you click a link to another page on that same site, will the cookie be automatically sent with the request? Yes. What if an attacker sends you a link to that same website in a malicious email and you click that link. Will the cookie be sent? Also yes." Then he says, finally: "What if an attacker directs you to a malicious website and upon visiting it your browser makes a post request to the original website that set the cookie? Will that cookie still be sent with the request?" And he says, "Yes!" So there are ways that this can be abused.

He says: "Cookies just don't care about how the request was initiated, nor from which origin. All they care about is that they are valid for the requested resource." He says: "'Origin' is a key word here, too. Those last two examples above are cross-origin requests in that they were initiated from origins other than the original website that set the cookie. The problem is that opens up a rather nasty attack vector we know of as Cross-Site Request Forgery, or CSRF."

He says: "Way back in 2010 I was writing about this as part of the OWASP Top 10 for ASP.NET series, and a near decade later, it is still a problem." So in his posting he gives some examples of how a POST providing both the old and a new password, for example, a POST which is in the response to a password change action, dialog and submission,

carries a promiscuous cookie which can be abused. I don't have it here because I can't describe it in detail in the podcast. But it's in the link that I provided, if anyone wants to see the details.

He explains that in a secure response, which is the second of the examples he offers, there are two anti-forgery tokens passed along with the request. One is in a cookie, and one is in the body, both of them called RequestVerificationToken. This is a familiar approach being used to deal with cross-site request forgery prevention, which is familiar to anybody who's been writing secure web applications. Many frameworks now just build this into the framework. It's a mess, but it's the best thing we have right now currently.

Anyway, so he talks about how both the cookie and the body carry a request verification token. He says they're not identical, but they're paired such that, when the server receives the request, it checks to see if both values exist and if they have been previously paired together. They belong together. If not, the request is rejected. He says: "This works because, while the one in the cookie will be automatically sent with the request regardless of its origin, in a forged request scenario the one in the body would need to be provided by the attacker, and they have no idea what the value should be. The browser's security model ensures there's no way for the attacker causing the victim's browser to visit the target site, generate the token in the HTML, then pull it out of the browser in a way that the malicious actor can access."

He says: "At least not without a cross-site scripting vulnerability, as well; and then that's a whole," he says, "that's a whole different class of vulnerability with different defenses." Anyway, the point is that we're living in a world where the use of cookies as our state management is vulnerable to exploits that clever hackers have come up with over time. So the change that Google has made in Chrome 80 to change the default to a way that blocks this class of cross-site scripting forgeries is expected to have some consequences.

So I salute Google for making the change. Microsoft plans to follow, although I think they're probably going to stand back a little bit and wait to see what happens. And Mozilla has indicated that it supports the idea also. In some reporting that I saw of this, it turns out that OpenID apps may be breaking. Microsoft warned very early on that these SameSite changes would break sites and applications that rely on OpenID-based federation.

Erik Anderson wrote, in a July 23rd Chromium Forum post on the use of cookies with SameSite, he said: "We love the intent and spirit of this change, but we fairly quickly determined that this breaks a large number of our sites leveraging Azure Active Directory (AAD) and Microsoft Account (MSA) authentication using the contract as defined here. We suspect that other OpenID-based federated auth providers may have similar scenarios and be broken."

Google warned IT professionals in its October 23rd Chromium blog post that there could be problems with internal applications and single sign-on implementations. They said: "Enterprise IT administrators may need to implement special policies to temporarily revert Chrome Browser to legacy behavior if some services such as single sign-on or internal applications are not ready for the February launch." And that was last week.

Microsoft has issued a specific warning about the coming SameSite changes. They said effects could be felt when using Microsoft Teams client applications. They wrote: "There are considerations for sites that use ASP.NET. Exchange Server, SharePoint Server, and Skype for Business client will all need to have the latest updates installed." So this is not something that has taken the industry by surprise. However, I wouldn't be surprised if it takes some people who haven't been paying attention or who haven't been keeping these applications updated.

So essentially people like Microsoft with ASP.NET, Exchange Server, SharePoint Server and so forth, they recognized this was happening, and they changed their technology to be compatible. For example, it may have been nothing more than saying explicitly SameSite=None, adding that attribute to cookies where they know that will not introduce a vulnerability. That allows the legacy third-party behavior to still continue, and it overrides the default if they didn't have SameSite equal something, which changed last week in Chrome.

Microsoft warned in a Microsoft Teams and SameSite cookies document that: "Applications running in the Teams desktop client are incompatible with the SameSite=None attribute, and they will not work as expected." They said the document offered a couple of workaround options. It also explained that the Secure attribute needs to be used when the SameSite attribute's value is set to None - that's that second thing that got changed here - to assure that third-party cookies won't get rejected.

For sites using ASP.NET or ASP.NET core, Microsoft warned in an October 18th ASP.NET blog post that the new SameSite changes will be in effect with .NET 4.7.2 and .NET Core 2.1 and above, and they could break OpenID Connect logins. Updates to .NET that were released back in December and November added support for the new SameSite behavior. So again, as long as everybody's platforms have been kept current and are current, this should have gone smoothly. I imagine people may already have discovered that things like authentication, among other things, are broken because, as we've talked about in the case of OpenID, you're bouncing the user around and deliberately using some of these browser features in order to maintain the connectivity of all these pieces. Unfortunately, that breaks unless it has been updated in the months preceding Google's change.

So we see another example here of security is hard, and change is even harder. But tightening up the default behavior of cross-site cookies will clearly be a good thing moving forward. And if we learned anything, it's that until there is some actual pain, some actual breakage, no one will change anything. So this is why I salute Google. They're biting the bullet. They're being willing to break a few eggs in order to force the changes that will in the end significantly improve the security of web applications for everyone.

So we end up with Chrome, you know, going first; applications, some things, some corner things, some edge cases probably getting broken. It's like, what? Why can't I authenticate anymore? Or why is my web app no longer working? I imagine the person will find out rather quickly what's going on. And then it's just, I mean, it's not like making the change is hard. If you're sure you're not going to introduce or perpetuate a security vulnerability, you just add SameSite=None in order to revert to what was the default behavior. But at the same time the reason that's been changed is to begin making some meaningful security improvements to cookies. This does that. And again, it makes cookies less promiscuous. And I salute Google for leading the way.

Leo: Yeah. I mean, I'm skeptical. I feel like they've got an alternative that'll work just as well for them and their advertisers. And so in a way this is just pulling, you know, we've got our method. You guys shouldn't use that other method. Sorry. We won't use that. Well, but it's better than nothing; right?

Steve: Yeah.

Leo: They have their own fingerprinting techniques.

Steve: Well, and the other browsers will be following suit, too. So Firefox recognizes that, you know, this is going to be a good...

Leo: The difference is Google's an ad company. Firefox is not an ad company.

Steve: Right.

Leo: So what I'm saying is Google's got its own means, and it doesn't really need third-party cookies. And, yeah, I'm skeptical.

Hey, good show. Thank you. Lots to talk about. You could find the show, along with SpinRite, the world's best hard drive and recovery utility. Hard drive maintenance, I left a word out, maintenance and recovery utility. It's GRC.com. That's Steve's home on the Internet. He has 16Kb audio for the bandwidth impaired, he has 64Kb audio for people with two ears, and he has a beautifully written transcript by Elaine, all at GRC.com. While you're there, not only SpinRite, but lots of free stuff like ShieldsUP! and all sorts of wonderful useful tools, Perfect Paper Passwords and on and on and on. GRC.com. He's @SGgrc on Twitter, if you want to follow him, and you can leave him messages there.

We have a copy of the show, too. We have audio and video at TWiT.tv/sn. And there's a YouTube channel. But, you know, honestly, the best thing to do is subscribe. Get your podcaster pointed toward our feed. Actually, if you search for TWiT on your podcast application, just press subscribe, subscribe, subscribe, subscribe, subscribe. Get all the shows, automatically downloaded. The minute you need them, you've got them on your stuff.

Steve, we will be back here next Tuesday, 1:30 Pacific. We should be on time next week. We were a little delayed by the Samsung event. 1:30 Pacific, 4:30 Eastern, 21:30 UTC. Live stream's at TWiT.tv/live. Chatroom at irc.twit.tv. Join the kids in the back of the class as they throw spitwads at us. And I will see you next week, right here.

Steve: Yup. We have President's Day holiday observation on Monday. And so you and I will be here the following day, on Tuesday.

Leo: See you then. Thanks, Steve.

Steve: Right-o, buddy. Bye.