
# URL "Ping" Tracking

**Description:** This week we discuss more news of Microsoft's Chromium-based Edge browser; the U.K. government's plan to legislate, police, and enforce online social media content; improvements to Windows 10's update management; news from the "spoofing biometrics" department; the worrisome state of Android mobile financial apps; an update on the NSA's Ghidra software reverse engineering tool suite; perhaps the dumbest thing Facebook has done yet (and by policy, not by mistake); an important change in Win10 1809 external storage caching policy; and a bit of miscellany and closing-the-loop feedback from our terrific listeners. Then we're going to take a close look at another capitulation in the (virtually lost) battle against tracking our behavior on the Internet with URL "ping" tracking.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-709.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-709-lq.mp3

---

SHOW TEASE: It's time for Security Now! with Steve Gibson. This week Steve's going to talk about the next evolution of click-tracking in browsers. There's the Online Harms White Paper. Microsoft Edge browser is now official on Chromium. There's the Galaxy S10's in-display fingerprint sensor - apparently it can be spoofed now - and so many more topics. I'm Jason Howell. I'm going to be here with Steve next on Security Now!.

JASON HOWELL: This is Security Now! with Steve Gibson, Episode 709, recorded Tuesday, April 9th, 2019: URL "Ping" Tracking.

It's time for Security Now!, the show where we talk about the latest security news happening this week. And of course, well, I'm Jason Howell. I don't sound anything like Leo. That's because I'm not him. He is enjoying himself on the beach, I think. But joining me as always, every single week, Steve Gibson, the man about town when it comes to security. How you doing, Steve?

**Steve Gibson:** Jason, great to be with you this week while Leo is basking in the sun on one of the Hawaiian islands somewhere.

JASON: I'm not jealous at all. Nope, nope, not going to be jealous.

**Steve:** I've done Hawaii a few times. I spent the second half of my honeymoon - the first half was in Napa, the second was in Hawaii. And it's like, I'm not really big on the super humid tropical clime. That's sort of not my thing. I like the dry air of air conditioning.

JASON: Okay.

**Steve:** So I'm sort of looking around because that's what I have right now.

JASON: You're surrounded by the dry air produced by all the technology and gadgets that are whirring in the background. I myself love a good tropical vacation. I have not been in literally years. This Thanksgiving we're going back to Hawaii. So we're going to get 10 days in Hawaii. And so I am very much looking forward to that. I can appreciate the dry air thing, but I need some humidity in my life. I need a break from all this dry air.

**Steve:** That's right. So we're going to talk about, when I began to assemble the news of the week, nothing really jumped out at me. But the more I looked into one of the topics, the more I thought, you know, this really does represent kind of an important drift, if nothing else, in the industry. There is a little known feature in URL links that we're all clicking on all the time, when we click on ads or we click on links in articles or just anything on the 'Net, which has languished for about well, let's see, at least a decade. And it was finally ratified in HTML5.

And the thing that popped this onto the news is that Google's Chrome browser, which we now know will be echoed, unless Microsoft makes changes, and it's unexpected, in their Edge browser, which is now Chromium based, it is removing the option to disable this. It was always enabled. It used to be disableable. Opera and Edge as it is today all have this enabled. Only Firefox and Brave don't. But it is sort of the ultimate in tracking our actions, sort of like the final capitulation to any hope of resisting the idea of being tracked on the Internet. It's like, no, that's the way the Internet is going to work, folks.

And so it's now supported officially, tracking is officially supported by the HTML protocol rather than being sort of a, you could argue, an unintended side effect, sort of in the way that third-party cookies - I've often talked about how they were never intended to be used to track us. Cookies were meant to allow persistent sessions with the sites we were visiting, not to allow third parties where we are not visiting to track us across the Internet. Similarly, the practice of redirecting our browser through redirect links was never intended to be used for tracking, but it has been subverted for that.

Well, this final move using an argument in URLs known as "ping," believe it or not, has no other purpose than for tracking. That's why it's there. And the fact that it is going to be always enabled, cannot be disabled - and Google is already using it. I show, and we'll get to it toward the end of the podcast, that when you bring up Google Search, you get two different pages if you are doing it in Chrome or in Firefox because Google is already using this, which they know is in their own browser because they put it there, and they know Firefox has it off by default. Anyway, I think that's going to be some interesting content for our listeners that they haven't come across.

But we've got much more than that to talk about. We've got some more news about Microsoft's Chromium-based Edge browser. We've got the U.K. government's plan to legislate, police, and enforce online social media content. Improvements to Windows 10 update management. News from the spoofing biometrics department. The worrisome state of, in this case it was Android mobile financial apps, but I explained it's probably all apps, it's just that the research was done on Android because it was easier for them to do. We've got an update on the NSA's Ghidra software reverse engineering tool suite. Perhaps the dumbest thing Facebook has done yet. And, you know...

JASON: Lot of competition.

**Steve:** I'm sure in the same way that this network has This Week in Google, you could do This Week in Dumb Facebook Things. But in this case it was done deliberately and by policy, not by anything they could claim as a mistake. So we have to talk about that because it's unbelievable. Also an important change in Windows 10 1809. They just finally ratified as of last week, we talked about that, update to Windows 10. They're changing in a significant way the default external storage caching policy. And I know that our listeners will be interested.

We also have a bit of miscellany; some closing-the-loop feedback. And then we're going to take a close look at this kind of sad capitulation is the best word I have for it, creeping trend toward making tracking official, rather than something that people could block if they were saying, I want to turn that off or please don't and so forth. It's just the way the Internet's going to be. So I think lots of fun stuff for our listeners.

JASON: It's just the way the Internet's going to be, says Steve Gibson. Basically what you're saying is it's time to put your hands in the air because tracking is here, and it's not going anywhere. It probably hasn't, you know, it's been here for such a long time that this just seems to be more of a cemented way, cemented approach.

**Steve:** It's also sort of like - I liken it to JavaScript, how a couple years ago, using NoScript on Firefox, if you were kind of an old curmudgeon, and you didn't like the idea of websites you're just randomly visiting running code on your browser, you could turn scripting off and then selectively turn it on if you needed it. Well, some time ago that just became impractical because websites, there were just like no websites left that would do anything without JavaScript. Mine, okay, except that, you know, that doesn't really count. So I don't have any script on my site except specific pages that you go to when you want - like the Password Haystacks page. It uses script to do an actual job for you, rather than endless tracking. And anyway, so yeah, we will get to all that. Going to have lots of stuff to talk about.

JASON: Absolutely, a ton of stuff to talk about. All right, security news time. Microsoft's Chromium Edge browser is now a thing.

**Steve:** Yes. With rather surprising speed this has become available. We covered the news last week of it having leaked, and there were kind of some sketchy places that you could download it. And I know I heard Mary Jo say that she was running it. What was interesting was it was running on Windows 7 as opposed to, as is everything now that Microsoft is doing, explicitly Windows 10 only. So I have the link in the show notes. Presumably you can google it also. But it's microsoftedgeinsider.com. I think that's probably all you need, www.microsoftedgeinsider with no punctuation there, dot com. There's a download page.

They declare that - so this is just to remind our listeners, or if someone has skipped a couple podcasts. Microsoft apparently threw in the towel - "capitulation" seems to be the word of the week on the podcast - just gave up all of the work that they had put into their Edge browser, which was arguably very good work, and said let's just use Chromium. Let's use the open source browser which of course Google's project Chromium has famously created, which is the number one browser on the Internet, despite all efforts of Microsoft's to make it hard for people to make that change. People want to use Chrome.

And I have to say, everywhere I go, I mean, I'm still holding onto Firefox. I like Firefox. It is arguably becoming - its differentiation from Chrome is increasing as Firefox continues to maintain more of an independent, non-ad supporting, non-tracking profile. And mostly I just like the UI. I mean, a browser is almost all about the user interface. And Firefox allows integrated vertical tabs down the left-hand side that I cannot live without. I mean, until I can't do that or until Chrome allows me to do that, and there's been some talk about it, I'm using Firefox because UI on a browser matters.

Anyway, so Microsoft gave up. They're switching to Chromium. The early reports from the leaked copy were all glowing. I mean, it was, you know, basically it's Chrome with an Edge wrapper around it from Microsoft. Well, it's gone official. It for Windows 10 is now available. And, interestingly, they intend to make it available officially for Windows 7, 8, 8.1, and macOS. So, I mean, again, I'm not that excited because I'm happy with Firefox

on my Win7. But if somebody for some reason didn't want to run actual Chrome, Google's Chrome on their Windows machine and wanted to have Edge, there is apparently some feature in Edge that would allow you to have IE tabs within the Chromium Edge browser, if presumably you went to some who-knows-what website that actually required IE, like real old Internet Explorer function that hadn't been updated to run on contemporary browsers, you could integrate that into Chrome.

And of course we also, a couple weeks ago, we talked about this bizarre add-on that Microsoft had officially created which was available both in the Google Chrome extension store and as a Firefox extension, which when you attempted to go out onto the Internet, would grab your browser and run it under Edge, if you were doing this in Windows. And it was like, okay, I'm not sure why that's a good thing. But anyway, they did that.

So there are, as this official Microsoft Chromium Edge site states, three update channels are available, which will be familiar to many people who enjoy playing on the bleeding edge. There's the Beta channel, which is the most stable browser. And that code is updated about every six weeks. Then there's the Dev channel, which will be more stable than the on-the-fly nightly updating Canary. But the Dev channel would be not yet ready for Beta. And it would be the one that's working toward the next major release. And the Canary channel is like the cutting-edge state of the browser, with code commits from that day, so it's updated nightly. And that's where prototyping of the newest features that would be two major versions away from the current release would be hashed out and experimented with.

So this is neat. I mean, this sort of feels like part of the new Microsoft, Microsoft beginning to do popular things the way other companies have been doing them and wanting to play with the rest of us also. Microsoft's Joe Belfiore stated in an announcement, he said: "In these first builds we are very much focused on the fundamentals and have not yet included a wide range of feature and language support that will come later." He said: "You'll start to see differences from the current Microsoft Edge, including subtle design finishes, support for a broader selection of extensions, and the ability to manage your sign-in profile. We look forward to people starting to kick the tires and will be refining the feature set over time based on the feedback we receive."

And then the one significant thing, and we saw this already in the leaked release, but it's still there in the official, is you can enable support for Google Chrome extensions in Microsoft Edge's insider build. You select Extensions under the browser's main menu, the Edge menu. And then down on the bottom left is "Allow extensions from other sources." If you turn that on, then you're able to use Google Chrome extensions in this Microsoft browser.

So I guess the sense is that I guess Microsoft is hoping that this will cause fewer people to choose Chrome over Edge if they are essentially the same. You know, I don't know. It seems to me like Chrome has such a powerfully strong positive grasp and reputation on the 'Net now that people just like the idea of not using, for some reason, I mean, I'm amazed. Everywhere I go I see that little Chrome logo in people's, you know, on their desktop or down in their quick launch area of Windows. And it's like, okay, well, it's obviously prevalent. So it'll be interesting to see in terms of share what happens.

There was a really interesting chart that I saw as part of this news that showed all of the Google-specific features which Microsoft had ripped out of their version of Chromium for Edge. I mean, it was like four columns of 20 items each. It was just pretty much everything. And again, naturally, Microsoft doesn't want to be promoting Google properties in their take on Chromium. But they really did have to, you know, there was a lot that is there in Chrome that will not be in Edge. And actually, maybe that sort of answers my question. If people want those things, they're not in the Edge version, they're in the Chrome version.

JASON: Do you think that this transition is the type of thing that people who follow tech might actually notice, but the everyday Microsoft Edge user is going to be completely oblivious to?

**Steve:** I agree. I don't think they would have any way of recognizing that that's what's going on. Certainly developers will care because suddenly Edge will be able to run Chromium extensions. In fact, it does run the work-in-progress SQRL login extension right now. So the SQRL project just got another major browser that it's compatible with. So that's an example of, probably more than anything, that may be one of the big benefits is that we know how incredibly popular browser extensions are. People want to customize their browser in all kinds of different ways. I mean, I run with a bunch of extensions in Firefox, like the one that gives me the sidebar of vertically oriented little tiny tabs so I can have 400 open.

JASON: Oh, you're one of those.

**Steve:** Uh-huh. Yeah, I have a scrollbar on my tab column.

JASON: How do you do that? I don't even understand. I don't understand people with that many tabs. It just breaks my brain.

**Steve:** Anyway, so it may be that that's the hook is that you'll be able to use that rich extension library in Edge. And so there was no way, you know, Microsoft tried to replicate that, and it just didn't happen. It's like Paul talking about the Windows Store and just how pathetic it is, just how awful the apps are there. Microsoft keeps trying to copy Apple with iTunes; and, oops, that didn't work. And here again they tried to create an extension experience for Edge, and no one cared. They just used Chrome because all the extensions that they wanted were there. And so Microsoft said, okay, fine, we'll just put Chrome inside our Edge wrapper, and everyone can use the extensions that they want, and maybe we won't lose so many of them.

JASON: Yeah. Well, and it's not without precedence, either. Microsoft did this also with its mobile strategy. Now basically we're getting Microsoft apps on Android instead of Microsoft controlling its own OS. So, yeah, this is just kind of the modern Microsoft era is realizing, it's kind of strange when they make this decision, but realizing that they're willing to kind of admit to a certain degree, whether they call it defeat or not, admit defeat in one area and shift their strategy in a new way like this.

**Steve:** Yeah. So yesterday the U.K. government announced a suite of online safety laws which, I mean, and this is what we saw coming. This is sort of like, you know, we know that the GDPR happened within the EU. So now the intent here in the U.K. is to hold the publishers of online social media platforms liable for the harmful behavior spreading through their platforms. And I guess my feeling is this is the inevitable legislative blowback from what has been the previous practice of providers, social media providers wanting to try to sell the idea that they're just providing a utility, and that they're not in any way responsible for the actions of the people who use this information utility that they're providing. And unfortunately, that strategy has, as we know, backfired a lot. This paper, which I think was 102 pages, and believe me I'm not going to drag our listeners through it...

JASON: Page one. Chapter two.

**Steve:** But for the next three months - and so this is not yet legislation. This is the slow grinding wheels of legislation beginning to turn. And so the first couple of teeth of those gears are meshing. So this is intended to be open and available for three months, until July 1st, so just a little less than three months. And to give our listeners some sense for

it, they said, sort of in the introduction, what they call the "Online Harms White Paper" sets out the government's plans for a "world-leading package of online safety measures that also supports innovation and a thriving digital economy." Uh-huh. "This package comprises legislative and non-legislative measures and will make companies more responsible for their users' safety online" - and of course we're going to march out the kids, so - "especially children and other vulnerable groups.

"The White Paper proposes establishing in law a new duty of care toward users, which will be overseen by an independent regulator. Companies will be held to account for tackling a comprehensive set of online harms ranging from illegal activity and content to behaviors which are harmful, but not necessarily illegal. This consultation aims to gather views" - that is, this paper, this thing that's out for the next 90 day - "aims to gather views on various aspects of the government's plans for regulation and tackling online harms, including" - and we've got five bullet points here - "the online services in scope of the regulatory framework," which is to say which online services would be in scope for this framework; "options for appointing an independent regulatory body to implement, oversee, and enforce the new regulatory framework; the enforcement powers of an independent regulatory body; potential redress mechanisms for online users; and measures to ensure regulation is targeted and proportionate for the industry."

So, and I pulled a bunch of points out. There were 49 numbered paragraphs in this 102-page thing. And I think I pretty much characterized it well enough. I mean, we know what its focus is. Paragraph 12 had some bullet points which I'll just quickly cover. They said: "Our vision is for a free, open, and secure Internet; freedom of expression online, yada yada" - it actually didn't say that - "an online environment where companies take effective steps to keep their users safe and where criminal, terrorist, and hostile foreign state activity is not left to contaminate the online space."

So basically we know what they're saying. They're saying we're going to make the likes of Facebook and Google and Twitter much more responsible for the content they're carrying than they have been to date. So things are going to change is the whole issue here. And so on. "Rules and norms for the Internet that discourage harmful behavior." They said: "The U.K. as a thriving digital economy, with a prosperous ecosystem of companies developing innovation in online safety; citizens who understand the risks of online activity, challenge unacceptable behaviors, and know how to access help if they experience harm online, with children receiving extra protection; a global coalition of countries all taking coordinated steps to keep their citizens safe online; and renewed public confidence and trust in online companies and services."

So this is the beginning of probably a reform of some sort for, which we've seen in various forms in the past, we've seen people talking about it, this is the beginning of something. And we'll certainly, to the degree that it affects the security and privacy space, we'll be talking about it, as I'm sure for years to come.

JASON: Yeah. And I'm always curious on something like this, or at least in the past short term with GDPR and all these other things that are happening overseas around privacy and protection and security, how this ultimately trickles down to places outside of the U.K. because that creates a very large, not impossible I imagine, but a large hurdle for technology companies to be heavily regulated in one area and not in the other. Maybe it's just easier for them to roll it out to everyone. You know what I mean?

**Steve:** Well, and Jason, you can't go anywhere now without acknowledging cookies. Everywhere you go, you have to say, yes, I know, okay, fine, I know.

JASON: True. Perfect point.

**Steve:** So that's hit the entire globe because of GDPR saying, you know, we're going to sue your pants off unless you let people know you use cookies. Well, everyone does. So it's like, what? It's like, look, we have a web page. Yes, fine, I clicked. I acknowledge you. I acknowledge you have a web page. That's why I'm here.

JASON: I acknowledged that when I hit Enter on the search query.

**Steve:** That's right. Oh.

JASON: Yeah. Yeah, it'll be interesting to see how this plays out. And as, like, I think the cookies example is the perfect example; right? Like through this, at least here in the U.S., we in essence have become somewhat desensitized to what that actually means, what that acknowledgment by clicking okay actually means. It makes it very easy for those regulations - we hear about regulations in the U.S. all the time now as relates to technology companies. It makes it even easier, less of a hurdle, for those efforts to roll out over here in a more official sense. So probably the same could be said about this, if and when it becomes what it is.

**Steve:** Well, and it's just cluttered our browsers with visual spam, the fact that you go to a site, and you have to acknowledge that they're using cookies. In fact, when I was researching this story, the U.K. government site put its banner at the top and had me click on it to acknowledge that it was improving my experience online. It's like, no, you're not. You just made me click this. That did not improve my experience online. Sorry about that. And the fact that I clicked it to acknowledge that it improved my experience, well, you gave me no choice.

JASON: No choice. You must like this. Click here to like this.

**Steve:** Yes.

JASON: Ah, yeah.

**Steve:** So I just, thank goodness, suffered through the Windows 10 1809 update this morning. It took several hours on this system that I'm talking to us on. That's why I'm saying I was like, I was holding my breath that it was going to finish in time for the podcast because it sat at 18% for, like, a long time. And I thought, okay, this better not be proportionate, or I'm not going to be talking to our listeners today. I mean, I would have scrambled around and set up Skype somewhere else, but that would not have been fun, either.

JASON: Do it from your phone.

**Steve:** Yeah. And as we said, we just last week, this very laboriously troubled - this was the October 2018 update, which was so fraught with problems that only last week did Microsoft finally decide that they had consensus from all parties, and sort of at this point permission, to try again to roll it out. So the point is that next month - so much time went by that it's supposed to be twice a year; right? Every six months? Well, next month, in May, is the next one. We're already there.

So what's interesting is that, in a very nice change, I think, they are going to what they're calling an improvement to the Windows 10 update experience with control, quality, and transparency. Last Thursday, on the 4th of April, Mike Fortin, who's VP of Windows, he said: "In previous Windows 10 feature update rollouts" - so those are the semiannual, right, the twice a year, not the monthly security fixes, but these are, like, oh, we got a whole bunch of new stuff for you. Whether you want it or not, we're going to make you take it, by the way.

But the feature rollout, he said: "The update installation was automatically initiated on a device once our data gave us confidence that devices would have a great" - great - "update experience." That's right. That's a great experience. "Beginning with the Windows 10 May 2019 Update" - so that's next month - "users will be more in control of initiating the feature OS update." So basically it's going to be on-demand to some degree. He said: "We will provide notification when an update is available and recommended based on our data, but it will be largely up to the user to initiate when the update occurs." But again, with limits, as we'll see.

"When Windows 10 devices are at, or will soon reach, end of service, Windows Update will continue to automatically initiate a feature update. Keeping machines supported and receiving monthly updates is critical to device security and ecosystem health. We are adding new features that will empower users with control and transparency around when updates are installed. In fact, all customers will now have the ability to explicitly choose if they want to update their device when they check for updates, or to pause updates for up to" - wait for it - "35 days." Okay. So not 365, 35. So, okay.

He says: "We're taking further steps to be confident in the quality of the May 2019 update." And you can imagine that's true after the fiasco of the previous October 2018 update. He says: "We will increase the amount of time that the May 2019 update spends in the release preview phase. We will work closely with ecosystem partners during this phase to proactively obtain more early feedback about this release. This will give us additional signals to detect issues before broader deployment." All of which they failed to do last time, of course. "We are also continuing to make significant new investments in machine learning technology" - so they're bringing AI in - "to both detect high-impact issues effectively at scale and further evolve how we intelligently select devices that will have a smooth update experience."

Okay. In the show notes I took a picture of the screen which they showed in this announcement. And this is what I like. There's the standard Windows Update, then the Check for Update button, and then there's a new section that stands off, sort of a pullout, and it says "Feature Update to Windows 10, version 1903," which is not yet, of course, but that's what it'll be. And it reads: "The next version of Windows is available with new features and security improvements. When you're ready for the update, select 'Download and install now.'" And then there's a link to do that. So I really like that.

So what this does is it removes it from the news appearing on your screen that, oh, we've just started a multi-hour process of giving you a bunch of new features you didn't ask for. Oh, and you can't use your machine in the meantime. They've switched it to "We've got new stuff available for you. Click here when you're ready to get it." Unfortunately, you still have to have it. It's not like they're making this entirely optional for a long period of time. Looks like it's up to a little over a month that you can postpone this, and also only when you go to manually check for updates.

So I guess this is intended for advanced users or update- and security-conscious users who do go to proactively see whether their system is up to date. Then they will receive the news that, oh, look, in this case the 1903 Windows 10 feature update is now deemed ready for this machine I'm using. I'll kind of keep that in mind for when I don't need to use the computer for a few hours, and then let her rip. So anyway, nice to see some improvements in, you know, like clearly Microsoft is understanding and listening to some of the feedback that they're getting about this because, I mean, I just did this. I just went to 1809. And like I said, I was terrified. It had the morning, basically, and it took it.

JASON: What is that rule that, like, no matter how much time you have, you always fill it with the amount of work that you have. Apparently Microsoft is adhering to that closely, as well.

**Steve:** That's right.

JASON: Yes, you were brave to do the update before the show. But you made it. That's the point.

**Steve:** Yeah, I did. I'm here.

JASON: You're here. All right. So you're going to talk a little bit about this little portion of the phone that drives me insane every single flipping time I need to unlock this phone.

**Steve:** Oh, interesting. I look forward to hearing your feedback. So as we know, and this is what you're talking about, is that Samsung's recently released Galaxy S10 top-of-the-line smartphone has maybe a not-so-spiffy new fingerprint reader. But at least it's kind of spiffy technology.

JASON: It's spiffy technology. I'll give it that.

**Steve:** Yeah. We all recall Apple's fingerprint reader. We talked about it at the time, couple years ago. It uses capacitive technology to image the ridges of our fingertips. When it was released it was noted that, since it was capacitive, it would not be spoofed by flat images. And as we all know, it was only a few days after it got into the hands of some creative hackers that a fake thumb with ridges, much like the Ruffles potato chips, was created to spoof Apple's capacitive technology. So not unspoofable after all.

So now we jump forward a few years to Samsung. Everyone these days is all freaked out over the seamlessness of their screens, with much angst like over the Apple iPhone ears, right, the little side things created by the encroachment of the iPhone's 3D facial recognition technology onto the screen. So Samsung, determined to minimize the encroachment of any UI technology, set their engineers the task of incorporating a fingerprint reader into the screen so that there would be no physical, set-aside, set-off zone of any kind. They succeeded and created an acoustic ultrasound imaging technology whose transducer is placed behind the screen and which operates through the screen. Apparently, based on Jason's experience, not as well as the engineers at Samsung were hoping.

JASON: Your mileage may vary.

**Steve:** I want to hear about that in a second. So we're talking about this because, as with Apple's capacitive 3D fingerprint imaging technology, it didn't take clever hackers long to spoof the Samsung ultrasonic fingerprint either. Last Wednesday a Reddit user calling himself darkshark9 posted his hack of his Samsung S10. I've got a link in the show notes to a video of him doing this. He unlocked his Samsung Galaxy S10 using his 3D printed fingerprint picked up from a photo of a wineglass taken using his own smartphone. However, to make his accomplishment more striking, he noted that the fingerprint image could be captured at greater distance using a DSLR camera to steal one's fingerprint further away than having to near focus.

And so, reading from what he wrote, he said: "I pulled the image into Photoshop and increased the contrast and created an alpha mask. I exported that over to 3DS Max and created a geometry displacement from the Photoshop image, which gave me a raised 3D model of every last detail of the fingerprint. I popped that model into the 3D printing software and began to print it." He says: "This was printed using an Anycubic Photon LCD resin printer, which is accurate down to about 10 microns," he said, "in Z height, 45 microns in x/y, which is more than enough detail to capture all the ridges in a fingerprint." He says: "It printed perfectly." And he said: "Print time was only around 13 minutes."

He said: "It took me three prints trying to get the right image height." And he said in parens, and this is typical, he says: "I forgot to mirror the fingerprint on the first one," meaning, you know, you've got to flip it over because what you're pushing against the screen is the reverse of the image that was taken when you look at the fingerprint that you've asked to be printed. But he said: "But, yeah, third time was the charm. The 3D print unlocks my phone, in some cases just as well as my actual finger does."

So where does this take us? My feeling is we could attempt to make the recognition process much more robust. So both fingerprint readers have been spoofed. We know that Samsung's face recognition was spoofed. Although it is more difficult to do, Apple's face ID has been spoofed with a fake 3D face model. So, yeah. These are not unspoofable technologies. We could go to further extremes to make the process more robust and less spoofable. In the case of a fingerprint reader, we could run a weak AC signal through the fingerprint tissue to determine its impedance. That would be much harder to spoof. All existing spoofs would fail that test. Or we could send several different frequencies of light into the finger and determine some measure of relative spectral tissue absorption. We could also watch that over a span of time, like a span of a few heartbeats, to detect capillary pulse using photo - I can't ever say this. Photo - I can't believe I can't say it. I can't say it.

JASON: My favorite part of the show, right now.

**Steve:** Plethysmography. Normally it runs off the tongue, but I need some more coffee. Anyway, so we know that Apple did go to great lengths to make their face ID largely immune to simple spoofs. You can't show it a flat photo because they've got infrared imaging to obtain a 3D model of the user's face, specifically so that showing them a photo, showing it a photo would not fool it. But capturing 3D images of people is no longer the stuff of science fiction. Consumer apps now do that regularly. So that's really not that big a problem. And we have 3D printers, as this existing hacker just noted. So getting 3D models is not difficult.

So I think that the right way to think of this is soberly and realistically. This is another example of the classic tradeoff of convenience for security. Are biometric systems as secure as non? And the answer is no, they're not. They are not as secure as a long password. But they are incredibly more convenient to use. And so the question you have to ask is are they secure enough for your application? And applications vary. Are they secure enough for a phone owned by an NSA person who's keeping secrets on it? No, they're not. Are they secure enough for your mom, who just wants to pick her phone up and answer it? Yeah, probably they are.

So unless the technology in our current biometrics is raised to the point where their cost starts becoming prohibitive, because doing all this extra stuff is going to cost more in a consumer setting, and it's going to make them more prone to refusing to accept that you are really you when you are because you're sad today, and so the facial recognition got tightened up, or you changed the glasses that you're wearing, and they no longer match the profile of what you had before, or who knows what? I know, for me, Touch ID doesn't work when my fingers are cold, oddly enough. So maybe there's a thermal sensor in there. I don't know.

The point is that the more we do to make them much less easily spoofable, the harder it's going to be to convince them that we are who we are, that it really is us. Inherently, when you reduce the chance of false positives, you also, I mean, sorry. When you reduce the chance of false, yeah, false positives, you also reduce the chance of true positives because you've so much tightened down the system that it becomes, I mean, the fundamental problem is that the signal, from a signal recognition standpoint, the signal we are trying to recognize is soft. And it's a fingerprint, which is inherently a soft signal. Things can happen to it. You can get cut. It can be a different temperature. You can have

scars on it. And it can be a spoof. And certainly a camera looking at a face is a soft signal, whereas a long password is an incredibly hard signal. It is not a soft signal. But soft signals, biometrics, are much easier to use.

So when I saw this, it's like, aha, I spoofed the Samsung S10. It's like, yeah, everybody has spoofed everything. If it's biometric, it is spoofable. I mean, but it is also, except maybe Jason in the case of your Samsung S10, much more convenient to use than a system that is less spoofable. So anyway, I just sort of wanted to take a moment to acknowledge that this whole domain of biometrics has the problem that it is convenient, but we are making a big tradeoff in terms of absolutely lockdown security. So tell us about your S10.

JASON: Well, I just want to add to that real quick what you're saying as far as biometrics. You know, increasing the accuracy means that we're less likely to get in at a convenient rate is essentially how I take that. One of the things that I've always wondered is, okay, so we've got face scanning. We've got iris scanning, although the S10 got rid of the iris scanner. We've got fingerprint sensors. So we've got multiple biometric aspects. Why not combine them so that you have kind of a combined effort of more likelihood that it knows it's you? Well, the face scan matches. So does the fingerprint. Therefore much more certain that this is the person who it claims to be.

But what you're saying there makes me realize why that's probably a bad idea because the more you do that, the more variables, the more kind of room there is for it to get it wrong. And the worst-case scenario would be that you could never get it 100% right, and you're locked out of your phone entirely because there's no way to get in because you can't get both of them to sync or to happen at the same time. So maybe that's not a good idea. But that always seemed to be, to me, the obvious direction is, well, they'll just combine them. I'm already used to putting my finger on the fingerprint scanner. And if I'm already looking at my phone, then use both of those data points and let me in. But it's probably much harder than that.

**Steve:** You know, for years the access to my datacenter, where I have a physical rack of servers, it used a biometric hand reader, which was looking at the physical size of the fingers of my hand. And so it had, like, pegs. And you'd stick your hand in it, and then it looked down on it and measured my finger sizes. And so you wonder, with all the technology that Apple has, and the fact that they've got this 3D multipoint scanner, if you couldn't show it your face and then also hold your hand up and let it measure your hand and decide if, you know, get like an additional factor for you.

JASON: Well, it's interesting that you mention that. The LG G8 ThinQ is a new phone that LG showed off. And it has vein detection, where you hold your hand over the front-facing camera. And so it's similar, in a way. It's not quite the same, but it actually authenticates around the veins in your hand and your wrist.

**Steve:** That's probably a pretty good signal because, I mean, that's going to be, like, very unique from one person to another.

JASON: Probably so. A little bit of social educating the users around not feeling ridiculous holding their hand over their phone. But, I mean, maybe that ends up being a more secure direction to go. I don't know. As for the fingerprint sensor on the S10, I would just say my frustration comes out of the fact that just on a daily use basis, because it's in the display, and it's kind of hidden down here, it takes a little bit of time. It's not immediate, the way I'm used to with like a Pixel, where I know exactly where my fingerprint goes.

**Steve:** Right.

JASON: It's just not as consistent as I'm used to. And so more often than not, I end up doing two, three, four touches and then entering my PIN to get in. And it's just so much time spent when it works the other way. And so it just got frustrating over time.

**Steve:** Do the apps draw a circle and show you where on the display the sensor is?

JASON: Yeah.

**Steve:** So at least you know where to put it?

JASON: Yeah, I mean, I'm trying to see if maybe - it's probably hard to see. Maybe in my single you can see it. But there's a little area down here that...

**Steve:** There is an indication.

JASON: There is an indication sphere. So that little fingerprint dot down there, that will appear when it's time for me to authenticate on an app, and the display is on. If the display is off, I don't know if I'll be able to get it - oh. It realized that I held it up, and it turned the display on. But if it's off, I should be able to do that.

**Steve:** That's interesting.

JASON: Of course I'm hitting it sideways with my thumb, so it's just it's doing a little haptic kick, but it's not actually unlocking. But if you know where it's at, you can get into it even when the display's off.

**Steve:** And so your sense is, I mean, if you had your choice, you would have had them leave the fingerprint reader where it was because you knew where it was, it worked, it was reliable, and it just, I mean, you know, it was less hassle.

JASON: After two weeks of using the S10 I still don't have the in-display sensor nailed. And I thought after two weeks of solid use I'd get used to exactly where my finger needs to go. I'd get better at scanning it so that it would let me in more often than not. And it's still not there. So I'm open to new technology 100%. But in my two weeks of usage it's been more headaches than it's been, oh, my goodness. You know what I mean? And so to that end, yes, I want a system that works better than that. That's how I feel about it.

**Steve:** And so the official pronunciation is photo plethysmography. I was practicing it. Photo plethysmography.

JASON: I saw your mouth moving. I thought you were talking to JammerB on the talkback. But you were pronouncing plethysmography. That's hilarious. That's great. Hey, but you got there. You got there.

**Steve:** Photo plethysmography, yes. There's one for the cocktail party.

JASON: No kidding. Say that five times fast.

**Steve:** So this is a bit of a self-serving study because it was contracted for by an Arxan Technologies that bills itself as the trusted provider of application protection solutions. So they're based in San Francisco. They contracted with another firm, the Aite [A-I-T-E] Group, to look at the security of financial apps on the Android Mobile platform. And so, not surprisingly, they're not good, that is, the security of these mobile financial apps is wanting. But the details, as always, are where there's some interest. Their PR, their press release, highlighted the results of this contracted-for analysis of a group of popular mobile financial apps. Their press release reported the discovery of widespread security

inadequacies and protection failures among consumer financial applications leading to the exposure of source code, sensitive data stored in apps, access to backend servers via APIs, and more.

So at the company that was contracted, the senior cybersecurity analyst is Alissa Knight. She's with this Aite Group who authored the study, which was titled: "In plain sight: The vulnerability epidemic in financial services mobile apps." And, you know, given what they found, I would tend to agree. There are a couple charts down below, a couple pages down, Jason, that you might want to put onscreen, which show some little itty-bitty pie chart diagrams, but you can get some sense for it.

So Alissa Knight examined the mobile apps of 30 financial institutions downloaded from the Google Play Store across eight financial sectors including retail banking, credit card, mobile payment, cryptocurrency, HSA, retail brokerage, health insurance, and auto insurance. Using tools readily available on the Internet, and of course those now include the NSA's Ghidra that we'll be talking about next, "Knight found nearly all of the applications could easily be reverse engineered, allowing access to sensitive information stored inside the code, such as improperly stored personally identifiable information, account credentials, server-side file locations" - okay, server-side file locations - "API keys, live deployment, and QA URLs used by the developers for testing the apps." In other words, these things weren't even beginning to be securely designed, or designed to impede their abuse.

"The research highlights a systemic lack of application-appropriate protection such as application shielding, threat detection, encryption, and response technology across financial services. Analysis of the mobile financial industry applications highlighted major deficiencies in application design, including easily reverse engineered code that exposes serious vulnerabilities including in-app data storage; compromised data transmission due to weak encryption and insufficient transport layer protection; and malware injection and tampering."

The Aite Group's key research findings included lack of binary protection: 97% of all apps - which suggests that they found one that had it because that would be 3%, so that would be one of the 30 that they looked at - tested lacked binary code protection, so one of them had it, making it possible to reverse engineer or decompile the apps exposing source code to analysis and tampering. Unintended data leakage: 90% of the apps tested - so that suggests that three out of the 30 had it, 27 didn't - shared services with other applications on the device, leaving data from the financial institution's app accessible to any other application on the device. Whoops.

Insecure data storage: 83% of the apps tested insecurely stored data outside of the app's control, for example, in the device's local file system, external storage, and copied data to the clipboard allowing shared access with other apps, and exposed a new attack surface via APIs. In 80% of the case they found weak encryption, that is, weak implementation of encryption algorithms or incorrect implementation of a strong cipher, allowing adversaries to decrypt sensitive data and manipulate or steal it as required. And 70% of the apps used an insecure random number generator, making the values produced easily guessed and hackable. And of course the need for high-quality entropy in applications is something we have talked about many times.

Alissa Knight, who did this analysis, wrote: "During this research, it took me 8.5 minutes on average to crack into an application and begin to freely read the underlying code, identify APIs, read filenames, access sensitive data, and more. With financial institutions holding such sensitive financial and personal data and operating in such stringent regulatory environments," she wrote, "it is shocking to see just how many of their applications lack basic secure coding practices and application security protections. The large number of vulnerabilities exposed from decompiling these applications poses a

direct threat to financial institutions and their customers. These resulting threats ranged from account takeovers, credit application fraud, synthetic identity fraud, identity theft, and more."

She said: "It's clear from the findings that the industry needs to address the vulnerable epidemic throughout its mobile apps and employ a defense-in-depth approach to securing mobile applications starting with application protection, threat detection, and encryption capabilities implemented at the code level. Of all the findings, the most shocking was without a doubt," she writes, "the SQL queries exposing information on the backend databases hardcoded into the app, along with the private keys being stored unencrypted in different subdirectories."

So, yikes. In other words, these readily reverse engineerable apps are using readily discoverable private keys to access sensitive financial institution backend databases via poorly encrypted SQL query credentials. So we don't know anything about the security or accessibility of the backend databases. But if the same developers who implemented the frontend had anything to do with the backend, it doesn't look good. We've got no reason to believe at this point that the companion iOS versions of these apps are not every bit as poorly designed. As we know, iOS is more tightly closed, but it's now standard practice for a common codebase to be used across multiple target platforms with only the UI customized for the particular platform.

On the other hand, if the attacks are against financial institution backend databases and services, going through the Android version of the app would likely be quicker and easier since, as we know, Android is deliberately open, which is one of its benefits for those people who would prefer to use an open source platform. And the fact now that this knowledge of the vulnerability of these apps, although they were not specifically revealed, but it's not hard to guess what 30 of them might be on the Android platform, that's worrisome because now the bad guys know that these things are poorly designed and easily reverse engineered. That's now public knowledge. And thanks to the NSA's release of Ghidra, the bar to reverse engineering them has not simply been lowered, it's been dropped on the floor. So reverse engineering this binary code is going to become a hobby for hackers.

So I have in the show notes the two charts that demonstrate the lack of binary protection that was found, the prevalence of insecure storage, unintended data leakage, client-side injection, weak encryption, implicit trust of all certificates. Oh, my goodness. That's like...

JASON: Nothing.

**Steve:** Like nothing. Execution of activities as root; world readable and writable files and directories is, like, prevalent; private key exposure, same thing; exposure of database parameters and SQL queries; insecure random number generator is a big problem. So, yikes.

And then on the heels of that we have last Thursday's widely anticipated release of the full source code for the NSA's Ghidra reverse engineering tour de force application which appeared on GitHub. I've got the link in the show notes to that on GitHub to remind our listeners, although I'm sure anybody, I mean, we've talked about this extensively because it is exciting, really, for the white and the black hat hacker community. It's written in Java. It's got cross-platform Windows, Mac, and Linux support.

It has incredible processor support: x86 16-bit, 32-bit, and 64-bit code; ARM and AARCH64; PowerPC both 32- and 64-bit; VLE; 16-, 32-, and 64-bit MIPS support; micro; 68000 in all varieties; Java/DEX bytecode reverse engineering; PA-RISC; the PIC family of microcontrollers 12/16/17/18/24-bit; SPARC 32 and 64; the CR16C; the Zilog Z80 instruction set; the 6502 that the Commodore 64 and the Apple II used; the 8051, which

is very popular in industrial controls; my favorite, the MSP430, the little TI microcontroller; the AVR8, the AVR32, and more.

So, wow. Like anything you could imagine, any IoT device, mainstream desktop software, this thing will reverse engineer it, provide a flow of control diagram, identify the APIs that the app is calling to. Those allow you, knowing what the APIs are that section of code is calling gives you a sense for what that region of code is doing. You can then begin to interactively, because the whole thing is an interactive disassembler, or almost a decompiler, allow you to figure out, as for example this person did, of these Android apps, exactly how a piece of code is working. The official site is https://ghidra-sre.org - that "-sre" is Software Reverse Engineering - dot org. So ghidra-sre.org. And, yeah, I think we're going to end up seeing some consequences of this.

Of course the tool that used to be quoted was IDA. That was the Interactive Disassembler, which cost more than thousands of dollars per seat. And I have a feeling that those guys are going to have a hard time selling IDA anymore. I mean, probably upgrades and feature improvements for people who already know it, what we hear of Ghidra is that it is - there's a steep learning curve, although there's tutorials and videos and all kinds of support available online. I have a link in the show notes to a threefold keyboard short keys guide. I mean, it's the full package. So I think probably in the future that's what the bad guys are going to be using because why not?

Oh, and it already has its first bug, which thanks to its wide exposure was found by a guy, a Matthew Hickey, who goes by the handle HackerFantastic. He reported a security issue when it was first released. He noticed that Ghidra opens debug port 18001, but it was bound to all of the system's network interfaces. That's a mistake. It should just have been bound to the system's localhost port. When a user launches Ghidra in its debug mode, that port will be open; and that would have allowed anybody with access to the machine over the Internet to remotely execute arbitrary code on the analyst's system. Whoops. So that got fixed. So that's the benefit. The NSA just got a benefit from their posting this thing on GitHub and making it widely available.

So again, props to the NSA. I've not had a chance to play with it. My world doesn't require much reverse engineering. I'm normally doing forward engineering, as our listeners know. But I think we're going to be seeing the fruits of much more readily available reverse engineering in the future.

JASON: But you know what will never change? Facebook.

**Steve:** Oh, goodness.

JASON: At least the story around Facebook that we're so used to.

**Steve:** Did you believe this, Jason? Oh. Okay. So I actually had to go back and make sure this wasn't posted on April 1st.

JASON: Right.

**Steve:** I mean, it's so bad. And even though it's of Facebook, I can still hardly believe that I'm saying this. Facebook last week was actually asking some of its users, I mean, with a straight face, to verify their identity by providing their private email address password. I'm not kidding. I have a screenshot. It's actually the Picture of the Week at the top of the show notes, but I have it also here, a screenshot of this, which has been confirmed. Daily Beast wrote about it. It was picked up, of course, by the tech press. Facebook has confirmed it.

So users at some point, while using Facebook, this was an interstitial that would pop up. It says "Confirm your email address." Okay? It says "To continue using Facebook," I mean, I can't believe this is not malware. "To continue using Facebook, you'll need to confirm your email address. Because you signed up with" - and this is blotted out in the screenshot because it was a real email address at, in this case it was gmx.net was the email provider. You can do this automatically through gmx.net. And then it shows in this form your email is already there, whatever it is at gmx.net. And then the next field says email password and a blank where the user is being asked to give Facebook their email password, which is none of Facebook's business. And nobody else has ever asked this, as far as I know, anywhere. But why not?

JASON: But don't worry. They're not going to hold onto it. Don't worry.

**Steve:** That's right.

JASON: So you can trust us.

**Steve:** And so you fill that in, and then there's a button, "Connect to gmx.net." In other words, instructing Facebook to log in as you to your gmx.net in this case, email to make sure that the password you gave Facebook works. This is not that I cannot pronounce plethysmography. This is that I'm still - I'm speechless in this case. It's unbelievable.

JASON: Yeah, and I mean, and somebody in chat actually poses the question that would be the alternative to this. Is it [chops], I think, in chat? Says why would they simply not send a message to the email account you gave them and have you confirm that you received it? Which is what we see nine times out of 10 anyway. That's what everybody does.

**Steve:** As everybody else in the world does. Yes. It's like, you know, please go to your email. You will find a link from us. Click the link to confirm that you are the owner of this email account.

JASON: Right.

**Steve:** It's like...

JASON: It's so weird. It's so weird. The thinking is so bizarre.

**Steve:** You have to wonder, at Facebook, what process could a company of their size have that would allow, I mean, it's one thing like for some idiot to, like a developer person, to propose this. But for like his boss not to say, "What?"

JASON: Wait a minute. You don't do that.

**Steve:** Like, "What?"

JASON: There's somebody on staff that must have thought that. Wait, wait, you're not supposed to do this. This is what malware does.

**Steve:** The Daily Beast wrote: "Just two weeks after admitting it stored hundreds of millions of its users' own passwords insecurely" - remember that I didn't even cover it on the podcast because it was like, okay, why? Who can keep up with all of the shenanigans of Facebook? But we learned that Facebook employees were able to look at their users' Facebook logon passwords that were stored unencrypted at Facebook. Anyway, Facebook, two weeks after admitting it stored hundreds of millions of its users' own passwords insecurely, "Facebook is demanding" - and this is Daily Beast's words, it's a

little strong in my opinion, "demanding," but okay - "some users fork over the password for their outside email account as the price of admission to the social network."

Daily Beast says: "Facebook users are being interrupted by an interstitial demanding they provide the password for the email account they gave to Facebook when signing up." And then they quote what I just read from the dialogue. And then: "Small print below the password field promises 'Facebook won't store your password.' But the company," writes the Daily Beast, "has recently been criticized for repurposing information it originally acquired for 'security reasons.' In a statement emailed to The Daily Beast after this story was published, Facebook reiterated its claim it doesn't store the email passwords. But the company also announced it will end the practice altogether." Oh, gee, how nice.

Facebook wrote: "We understand the password verification object" - I'm sorry, I even have a hard time saying it. "We understand the password verification option isn't the best way to go about this, so we are going to stop offering it." Oh, we're going to stop offering you the option to tell us your email password.

JASON: Gee. Thanks. So nice of you.

**Steve:** Un-effing-believable.

JASON: But not at all surprising.

**Steve:** Oh, lord.

JASON: All right. What do we have in store for us next?

**Steve:** So this was just sort of on the techie edge, but it crossed my radar, and I wanted to share it with our listeners because, again, I'm sure there will be a bunch of people here who care. With this next update, the feature release, the May 1809 - wait, no. No, no. The one we just had, the one last week is the October 1809. The one I just spent the morning hoping would complete in time for the podcast. Next month's is 1903, I think it was. Anyway, so 1809 is the most recent feature release. Microsoft disclosed a change in the default removal policy for external storage media plugged into any computer running this most recent version.

In their summary they said: "Windows defines two policies, Quick Removal and Better Performance" - hmm, which would I prefer? - "defines two policies, Quick Removal and Better Performance, that control how the system interacts with external storage devices such as USB thumb drives or Thunderbolt-enabled external drives." And here it is. "Beginning in Windows 10 version 1809, the default policy is Quick Removal. In earlier versions of Windows the default policy was Better Performance." They note: "You can change the policy settings for each external device, and the policy that you set remains in effect if you disconnect the device, then connect it again to the same port on the computer."

Okay. So translation. Historically, Windows has always enabled write caching for all mass storage, both internal and external. This is especially crucial for external nonvolatile thumb storage drives for two reasons. First, they are so very slow to write. They often read quickly, but they've got notoriously slow writing performance. And second, writing, as we know, inherently degrades the storage cells of nonvolatile memory and shortens the overall device life.

Okay. It turns out that users were not reliably using the Safe Eject function for their external USB storage drives. They were simply pulling the drives out of the machine before the drive's writing of cached data had been fully flushed and written to the drive.

So after all these years, Windows has decided to err in the direction of caution because apparently this was causing lots of problems with data corruption on thumb drives and scary notices popping up on the screen saying you pulled your thumb drive out too early. They decided to protect users from themselves by disabling USB device write caching by default.

There are several problems with this, in my opinion. First of all, I get it that, yes, I can see where Microsoft would err in this direction. I'm hoping that somehow users will be notified that they have an option to change this. People who are responsible and who understand what's going on get many benefits from write caching to external storage. So first of all, two kinds of caching, read caching and write caching. Read caching keeps what has been read from a slower device, whether internal or external, in RAM, so that if it's being asked for again, it's already there.

A perfect example are the allocation bitmaps for clusters on the drive. There are bitmaps that store bits which indicate whether clusters are available or not. While a file is being written, and new clusters need to be allocated, the system is constantly reading from that bitmap. And as it allocates sectors, it's flipping bits. It's setting bits in the bitmaps, saying these clusters are now no longer free for use. So having them in memory speeds up the checking, the reading, substantially.

But here's the other thing, is allowing that cached memory to be write cached means that all of those individual writes setting bits are allowed to accrue. And only after that region has quieted down, has calmed down, will that bitmap finally be written once out to the drive, whether it's internal or external, if write caching is enabled for the external drive. So not having write caching on an external drive which is relatively write, not only slow, but intolerant, meaning it hurts it to write to it, it is shortening its life.

So Microsoft unfortunately felt the need to flip the default. And I can understand it because scary dialogs popping up and corrupting external drives, that's a problem. But for we who listen to this podcast and who are able, who know to right-click on the drive in Windows Explorer and click Eject and then wait for the notice that it is now safe to eject this drive, the reason that is all happening is that any pending writes are being flushed to the drive so that the cache is no longer "dirty," as the jargon goes, in our machine. It's all been pushed out to the drive, which has accepted it, and it is now safe to pull it off the USB port

So the good news is there is the option, there's a Policy tab on the drive's properties. So under the Disk Management app in Windows, you can right-click on the computer. You'll find Management. Go to Management. Then go to Disk Management. That'll show you a bunch of drives. You right-click on one of them, do Properties. That pops up a dialog with a bunch of tabs. The Policies tab, the first thing there, it's called the Removal Policy. And it has Quick Removal, which is now the default. And it says, "Disables write caching on the device and in Windows, but you can disconnect the device safely without using the Safely Remove Hardware notification icon."

The unselected thing is Better Performance, which says, "Enables write caching in Windows, but you must use the Safely Remove Hardware notification icon to disconnect the device safely." And then below it is write caching policy. It's grayed out completely if you are asking for Quick Removal because there is no write caching. If you select Better Performance, then the write caching policy options light up, and you have the option of enabling - it says, "Enable write caching on the device. Improves system performance by enabling write caching on the device, but a power outage or equipment failure of removing it prematurely might result in data loss or corruption." And then underneath that: "If 'Enable write caching on the device' is enabled, then you have the secondary option of turn off Windows write cache buffer flushing on the device."

And it says: "To prevent data loss, do not select this checkbox unless the device has a separate power supply that allows the device to flush its buffer in case of power failure." So there they're saying we're going to allow write caching to be additionally "lazy," as the term is, which will give you further performance boost and further life extension of any USB devices that you have plugged in at the cost of then being really important that you don't yank the USB device out because basically you've said I will take responsibility for notifying Windows when I'm going to remove this thing because, even with write caching on, but they didn't say it, but the write cache buffer flushing off, you're allowing the device, the write cache to remain dirty permanently until you shut the system down. So again, additional performance, but you really then are taking responsibility for getting the data written to the drive. Again, you get better performance, but at that price.

So anyway, I just wanted to bring our listeners up to speed on that. This just changed with the most recent release of Windows 10. And so consider which storage you're using, how you're using it, and where you want to be responsible. And for the common user, again, reading from USB is not a problem. This is all about writing back to it. It's better if you take responsibility, but some people can't do that.

A couple bits of miscellany. I have a tip. As I was saying to Jason before we began recording, I thought I knew my way around Windows pretty well. I ran across a thing, a tip, a shortcut, a feature that I really want. And I used to install a third-party thing in order to get it. So Brad Silverberg, who was once upon a time a neat VP at Microsoft, I had a great relationship with him in the early days of Microsoft. We worked together on a number of things. I mean, it was a great working relationship. He left a long time ago. He's one of the good Brads.

One of the things he told me when we were talking about Windows is that he felt that one of the most underappreciated features of Windows Explorer was the ability to use "copy" on files. That is, we're all familiar with copy and paste of text within and between documents. But one of the ways to, for example, move a file around in Windows or between drives is drag and drop; right? You select it, or maybe multiple, by using Ctrl to toggle selections, or Shift to grab a range. And then you will left-click drag, or in some cases right-click drag, if you want to move them all somewhere else on the drive or move them to a different drive. He was noting that, and it is certainly the case, that there are instances where the drag-and-drop style is inconvenient. For example, especially later in Explorer, sometimes you have to maneuver or navigate to a different area of the file system, and you'll lose your highlighted selection, or it'll be off the screen or something.

Well, Brad noted something that I had used a lot, is you can right-click on that file or files and, for example, select Copy. And nothing happens except that the system remembers that set of files that you have kind of like pre-copied. And then you go somewhere else, select a drive, right-click on it, or select a directory, right-click on it, and click Paste. And it performs the equivalent operation of having done a drag-and-drop. The point was, he thinks that's really cool. He really felt that was underappreciated. And so I assume our listeners knew about it. But if not, there's a really cool tip.

But on my Windows XP machine which died, I had a - I think it was a tool from Microsoft in the old days that was Send As. It was an add-on to the Properties menu of files that allowed you to "Send to clipboard as." And that's, yeah, that's what it was. And so you could send things to the clipboard "as." The one I used all the time was like, I would right-click on a filename that was a long filename with spaces and hyphenations and nightmares, way down some gnarly path in Windows, and I would right-click on it and say, "Send to clipboard as name." And what that would do is it would paste that, the entire path and filename, as text onto the clipboard, which is super handy because you could then paste it into email. If you're a command prompt guy like I am, you could do Alt Space and then EP for Edit and Paste, and it would appear, it would paste it into wherever the cursor was on the command line and so forth.

Anyway, as our listeners know, my Windows XP machine died. I'm now on Windows 7. Yesterday I had exactly this problem, and I had not yet dug up that "Copy to clipboard as" widget. It turns out it's built in. It is built into Windows. It's always been there, and I never knew about it. And here it is. If you right-click on a file, you get the standard list of things, you know, Cut, Copy, Paste, whatever, Properties and so forth. If you hold the Shift key down when you right-click, one additional item is added to that context menu, "Copy as path." And so rather than copying the file, the way Brad talked about as being a cool way of getting ready to paste it somewhere, or maybe in some cases you could even use Cut in order to do a move, if you hold Shift key down when you right-click, you get one more item on there, "Copy as path." And so my world is complete.

JASON: Why do we have to work so hard for that one extra thing to appear in the context menu?

**Steve:** Wouldn't that be nice, just for it to have it there?

JASON: It's just another item. I don't know.

**Steve:** I know, I know. I agree. But now our listeners know that they can hold Shift down when they right-click and grab, you know, copy the entire...

JASON: At least it's possible.

**Steve:** ...path name of the file. That's like, yay.

JASON: Yes.

**Steve:** So I have shared a tip. And I'm looking at time and what we have left to do. So let's talk about built-in click tracking going mainstream. As I teased at the top of the show, this came to my attention because the current version of Chrome 73 has this enabled, but can be disabled. That "can be disabled" is no longer true in the staged but not yet released 74, and in the much more raw Canary 75. We're losing the ability to turn this thing off. Firefox has it off. It can be turned on. Brave, which is the privacy-focused browser, as we know, doesn't even allow you to turn it on. It's not an option.

Okay. So here's what's going on. In the HTML tag, probably the very first, okay, well, I guess there had to be HTML and title and body. Those tags had to exist in order to create a page. But the other one was the <a> tag; right? Just the open angle bracket "a," then you say space, href=, quotes, and then a URL, close quotes, then you close the angle bracket, and then you have the name of the link, and then you close the enclosing <a> tag, and that's - you just created an HTML link which, when you click it, takes you to where - jumps your browser to where that URL is referred to. That's the whole hypertext, you know, web that was originally envisioned by Tim Berners-Lee. So anyone who's done any HTML coding knows about that.

It turns out that there are many options that can also be added to that <a> tag, inside that <a> tag, in addition to that href. If we click on the link, what we're really telling the browser is reload this page with the contents of the URL at that domain that we're provided by the server at that domain. So if we're clicking on links on the same site, well, then the server knows that's what we're clicking on. I mean, it knows that we're moving around its site. If we click on an offsite link, what we're doing is we're telling our browser, reload this page with the contents of a page on a different server, from a different domain. And essentially we are leaving that site. So we might suggest that it's not any business of the site we're on where we went. And in fact the site we're on would not know. I mean, because we're on our browser, on text that our browser has loaded,

and we're clicking on a link that takes us to a different domain. So we have the right to do that.

But of course Google, for example, would disagree strongly with that philosophy. When Google brings us, presents us with a big list of search results, they are very interested to know which link we click on of those results as we leave their site. I mean, the whole point is we are leaving Google. We're going somewhere else. Thank you very much, Google, for the list of search results. We came to you because we want to leave and go somewhere else that you found for us, thank you.

But of course Google has evolved a lot since those early days. So they want to know - and they could argue they want to know because it will help them present better search results if they know which links people like from what their automated systems produce. But as I said, things have changed a lot. Now Google wants to know who we are based on which link we chose and build a profile of us.

Okay. So in the old days the traditional way of doing this was with what was called an HTTP redirect. And if in Firefox you right-click on a Google link on Google search results and copy the links, you know, the link address, where the link points to, and paste it into Notepad, you will see that the link does not point to the target site.

In fact, what you see in Google's results is fake. That's not where the link points. It's Google's results are not showing you the truth of those links. You right-click on a link to copy the link address, paste that into Notebook, what you get is something entirely different. It refers to www.google.com, and that link's actual destination is way back somewhere in the URL's tail. So in this fashion a link clicked in Firefox first jumps the browser back to Google, passing it the address of the actual target so that Google, after recording that fact and the user's browser cookie, et cetera, for tracking purposes and, as I said, to presumably maybe make their future search results better, and to certainly learn more about you, then Google redirects the user to the target site. This is the way it's always been.

And you were putting up on the screen, and I'll just mention that, for those listening, in the show notes I captured the actual URL from a Google link. Last night I put "oh my god" into Google, and I got a bunch of searches. One apparently is a movie from 2009, and so I just right-clicked on the IMDB link for a movie called "Oh My God" in order to see what came up. This is the link. And sure enough, it is - oh, I'm sorry, no, I did this all in Firefox. And sure enough, the <a href=", and so it starts with /url?. Okay, so of course the fact that there's no domain there, the backslash means it's going to stay on the current site. So this link is to Google, providing a bunch of gibberish stuff, all kinds of who knows what, I mean, it's unreadable nonsense that Google understands because it produced it in the first place. And then Google will end up sending you, your Firefox browser, on to this IMDB page.

So what's interesting is that the behavior is entirely different if you right-click on the same link in Chrome. Do the same search, same link in Chrome. There you receive the actual target URL directly, to which the browser will be targeted. And I show that also in the show notes. In Chrome you do the same thing, you get <a href="https://www.imdb.com/title/ and then the title of the page. In other words, in Chrome it goes direct. How does Google know what link you clicked on? You know they're not going to let go of you unless they find out who you are and what you clicked on and where you went.

Well, the answer lies in the second argument to that URL, which is ping=" and there it is, /url? and then more gobbledygook. So this is using a little known parameter, long since been around, as I said, like at least a decade this has been kind of talked about. It's been around. It's been there. Never really formalized. It got formalized in HTML5. So

in addition to an <a>, the anchor is what <a> stands for, the anchor tag having an href, which is what it's always had that takes you somewhere, now it can, as of HTML5, also contain a ping parameter and data. And what does it do? Well, like the name says, it pings any, and I do mean any, other server on the Internet with data identifying you because it's a ping from you, so it will contain a cookie from your browser for that domain. And it is actually issuing a POST, an HTTP POST. You know how we have GET and POST. So it sends a POST with a length of four characters, P-I-N-G. And in the headers, the query headers to the POST, is additional data.

So why the difference? Because Google knows that Chrome, their own browser, supports native URL following ping tracking so that it doesn't need to bother with the time-consuming URL redirect dance. Google generates, as I noted, very different results pages when it's rendering for Chrome versus another browser, where it cannot rely on ping tracking being present. So it was at least 11 years ago I found a reference to it. And unlike URL redirection, which is I would say arguably being abused in a way similar, as I mentioned at the top of the show, in a way similar to how third-party cookies are being abused, where URL redirection itself has many non-tracking purposes, ping tracking has no other purpose than tracking. That's what it does. It sends a ping when someone clicks a link. And it sends a ping somewhere else when someone clicks a link.

So for many years the purists doing the web engine development held out and kept ping tracking marginalized and was not something that could be relied upon as being present. That's true even today over in Firefox. But as with so many other skirmishes, that war against tracking is being incrementally lost. In today's Chrome 73, as I mentioned, it's enabled by default, but it can be disabled. As of 74 and 75 and presumably hereafter, it's just there. It's part of the protocol, it's part of Chrome, and it'll be on. And one has to imagine that'll be the case with Microsoft's adoption of Chromium. Apple's Safari also has it stuck on, as does Opera. So at the moment, Firefox and Brave are the two holdouts. They both have it disabled by default. Firefox does have the option to enable it. Oh, and get a load of this. The parameter list for ping is actually space-separated URLs. So not even one, but multiples are allowed.

So essentially this - oh, I was confused by my notes. I said that the POST contains the single word "ping" because I have the details here. Its query parameters provide Ping-From and Ping-To and the Content-Type as text/ping. So it is made for tracking. That's its purpose. So what I found most interesting is that there is also no same origin policy control over ping destinations. The browser is not constrained to only ping back to the page's origin domain. The page can instruct the browser to ping anyone, anywhere that the page's code requests.

And at first blush, as I was thinking about this, this might seem irresponsible of the designers. But I'm sure those arguing for this flexibility noted that URL redirects can also jump to anywhere without restriction, so why artificially restrict the built-in facility that's being offered up as it is arguably cleaner, if a bit creepier, replacement, that is, in the URL ping argument.

The two things that it has going for it is that I do agree that it's cleaner, that is, the href shows the actual destination of the link, and then the ping shows the site or sites who are going to be notified when you click on that link. So it's a little more straightforward. The other thing, though, is - and I think this is the part of Google that is really wanting to optimize and improve speed and improve security. And there's a lot of Google that is that way. This allows these processes to be done in parallel, which I would argue makes it worthwhile. A redirect is inherently a serial process. In the case of clicking on a Google link, you are going to Google. You're staying on Google, pulling up, like making a query to Google that then logs whatever it does, figures out all of this gobbledy-gook in the URL, and then it gives your browser a 302 redirect to the destination, and you go there.

I would argue that doing this in parallel probably gives the user a snappier response because their browser is immediately jumping to the site given in the link. And the documentation makes it clear that the ping is done asynchronously in parallel, thus not delaying their arrival at the new site by bouncing them among one or more tracking sites first. I don't know if anyone has ever noticed their browser's URL address field when this redirection chain gets really ridiculous, you'll see it flashing with all kinds of nonsense. Sometimes that occurs. You'll notice it like when a site is unresponsive or slow, your browser has a chance to update the URL field, and it's like, what the heck? And then you may, if you're lucky, finally get to where you're going. Well, that obviously delays you getting to where you want to go.

And so the use of ping instead of this at least allows this to be done in parallel and doesn't allow any of the redirection sites to waylay you if they would want to because of course your browser is turning, as it jumps from one of these to the next in a redirect chain, is turning control over to that site. It doesn't have to forward you on to the destination if it didn't want to. So this is cleaner from that standpoint, and probably faster. So I don't know when Firefox and Brave will finally capitulate. I think that they might as well do so, frankly, although it would be sorry to see it happen, because redirection chains work to solve the same problem, and they cannot be blocked without breaking lots of other useful things.

So I don't like the idea of making tracking explicit like this. It seems like we've lost another bit of our online freedom. But as I said, it reminds me of the way things once were with JavaScript, where it was practical to turn it off and turn it on only on sites that needed it. Now you really can't use the web without JavaScript running code on your browser. So I think similarly there is no practical means of blocking tracking unless you really, really want to spend your life trying to do so. I just think this is a battle that the modern web has forced us to lose.

And so those of us who consider ourselves purists are like, well, okay. I would argue we do get something in return, which is faster response because this tracking can now be done in parallel, not in series. At some point I think Firefox and Brave - maybe not Brave. Brave may just hold out, although in this case they're not really giving us any substantial privacy improvement because, as we've seen, Google issues a URL with redirection if it can't use ping. So we might as well just let it use ping and have our links followed more quickly with the ping happening asynchronously and in parallel in the background.

JASON: Hmm. It seems, I guess, what you just said there kind of answers the question that I had, which was just stripping out the ping entirely. But then it's going to fall back on a different tracking mechanism anyways.

**Steve:** Right, right.

JASON: Because it seems so easy to just do that. I'm sure that would be very easy, to just look for the ping, remove it, but there you go. All right. Have we missed anything? I'm digging into the cracks to see if there's any new insecurities.

**Steve:** We're at two hours, and that's our goal and our show.

JASON: I think we did it. Right on. Steve, wonderful stuff here. You know all about GRC.com. Well, if you don't, you should know all about GRC.com. That's where you can find everything that Steve is working on. Of course SpinRite, which he didn't get a chance to talk about today, but SpinRite, the best hard drive recovery and maintenance tool. You can get your copy there. Information about SQRL, updates, where you're at, where you're going with SQRL. People are eagerly following that, and you can find that at

GRC.com. Audio and video of this show, of course, transcripts of the show that you can only find there: GRC.com.

And then of course our website is TWiT.tv/sn. There you're going to find also audio and video of all the episodes of Security Now!, including this one. If you want to start, go back in time, you could do that. You can binge on all of the episodes back to back for literally weeks and weeks to get through them all.

**Steve:** A surprising number of our listeners do.

JASON: I'm not surprised.

**Steve:** And more recently we've just been doing pure news because, I mean, we've been doing two hours a week of pure news. There's so much to talk about. But back, I don't know why it was, I guess things were quieter once upon a time, but we really, we did some deep dives into the fundamentals of processor technologies, how the Internet works, I mean, there's a bunch of really good, timeless tutorial stuff back there. So I would commend our listeners to considering it, if they're looking around for something edifying to listen to, after they catch up on all the news of the week.

JASON: Absolutely. That's right. That's right. TWiT.tv/sn. And also you can subscribe there, of course. This is a podcast, so all the subscription details are there for audio and video. If you want to check us out live, this show records live every Tuesday starting at around 1:30 p.m. Pacific, 4:30 p.m. Eastern, 20:30 UTC, although that might be out of date. It might be different from that.

**Steve:** Maybe it's 19:30, I think.

JASON: 19:30, thank you.

**Steve:** I think it goes between 19 and 20, yeah.

JASON: I lose track. It's one of those. Just go to TWiT.tv/sn, you'll find the actual details. And I'm sorry I can't be more informational or solve that problem for you right now. But Steve, you've solved many problems for people on this episode. Really appreciate it. It's always fun doing a show with you. Thank you, Steve.

**Steve:** Thanks, Jason.

JASON: We'll see you.

**Steve:** Until next time with you.

JASON: That's right, until next time. But next week it'll be Leo returning on Security Now!.

**Steve:** Okay, buddy.

JASON: Take care, everybody.

**Steve:** Bye.

JASON: Bye.

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

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