I Know What You Did Last Summer

Description: This week we take some deeper dives into fewer topics. We look at a bunch of the new features offered by Chrome's latest update. We look into the fascinating details of a Russian attempt to co-opt and bribe an employee of Tesla, and at some sobering security research which successfully circumvents Visa's point-of-sale PIN protection, allowing purchases of any amount. We also have a bunch of closing-the-loop feedback and miscellany. Then we examine the surprising research into just how well knowing where our browser has gone in the past identifies who we are today. Knowing what someone did last summer tells us who they are with surprising accuracy.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-782.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-782-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We go on a deep dive in the new Chrome 85. We'll talk about problems with the credit card standard EMV. And then Steve's going to talk about a fingerprint technology that doesn't use cookies and is much more accurate. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 782, recorded Tuesday, September 1st, 2020: I Know What You Did Last Summer.

It's time for Security Now!, the show where we cover your security, your privacy, your safety online with this guy right here, the man in charge, Mr. Steve Gibson. Hello, Steve.

Steve Gibson: Yo, Leo.

Leo: Good to see you.

Steve: Great to be with you for this first day of September. Where has - well, I know where the year went. It's been quite a year.

Leo: I hope it goes a little faster. Let's move it right out. No need to delay.

Steve: Yeah, I would like to be in my time machine and jump these next two months up to the beginning of November.

Leo: I wouldn't mind jumping two years at this point.

Steve: But we've got to move through September and October first.

Leo: Yeah, yeah.

Steve: I wasn't sure when I titled this show, the title just popped into my head when I realized that the fun thing to talk about would be some updated research which was originally conducted, the first round, eight years ago in 2012, about the degree to which an analysis of our browser's history, just a static analysis of our browser's history can disambiguate users from each other. That is, where we've gone tells anyone looking who we are. So of course "I Know What You Did Last Summer" is the name of this podcast this week. And we're going to take some deeper dives into fewer topics this week. Sometimes we just quickly cover a bunch of stuff. I wanted to look at a bunch of the new features offered in Chrome's latest update...

Leo: Oh, good.

Steve: ...that we talked about last Tuesday, Chrome 85, just as it was coming out. And oddly enough, I had to go into About Chrome last night on a system where I've been using Chrome off and on since the last week, but it still hadn't kicked into its auto update for whatever reason. We're also going to take a look into the fascinating details of a recent, as in last week, culminating last week, but pretty much through the month of August, attempt by Russian-sourced individuals to co-opt and bribe an employee of Tesla. The details are interesting. Also some sobering security research which successful circumvents Visa's point-of-sale PIN protection, allowing purchases of any amount with a stolen Visa card and just completely bypassing the need for a PIN.

Leo: Wow.

Steve: Also we have a bunch of closing-the-loop feedback with our listeners, some of which is really neat, and some miscellany. And then we're going to, as I said, examine this - oh, not only was there research from eight years ago, but it was just updated by a trio of researchers at Mozilla a week ago. So a bunch of neat stuff to talk about. And we also update our Picture of the Week, which was humorous, and which we showed earlier this year, like I don't remember, like maybe late March or April. It showed a timeline of the way the COVID-19 pandemic had changed the importance of things, like a sudden jump in toilet paper and a gradual increase in the use of the Internet. A pretty straight line need for coffee, but alcohol consumption increased. Anyway, that picture has been updated kind of wonderfully.

Leo: Oh, yeah, that is one more item to really put it over the top.

Steve: It just, every time I look at it, Leo, it just cracks me up. It so perfectly does.

Leo: It's really - it's our life. Well, we know what you did last summer, and we know what we did last summer. We were here, busy. And boy, the security landscape didn't change things much. I'm going to have to make some room on the screen for this one.

Steve: So people are going to have to check out the show notes if they're curious to see this. Suffice to say this thing tracks from March through June, the "Relative Importance in 2020 So Far" was the title, of coffee, of your car, of the Internet, of shaving, of alcohol, of toilet paper,

and of sweatpants. And remember the need to shave dropped down because you didn't have to be going to work every day. There was this weird toilet paper shortage when suddenly everybody was going to be staying home and worried about how long they were going to be there, and so people over-purchased toilet paper, thus creating sort of a synthetic shortage of it for a while.

Of course Internet grew in importance. Cars dropped importance dramatically because people were not commuting. Sweatpants increased in importance because you didn't have to get dressed. And after all, no one can see you from the waist down when you're using a Zoom conference. You're like, no, I've got my pants on, yeah, uh-huh.

Anyway, the point is that a last item was added which was the relative importance of masks. And what's so funny about this is that it looks like the walk of a drunken ant on the paper. It's sort of up and down and goes around in circles a bit, and it does a peace symbol at some point, you know, it's just wonderful. So anyway, I thank a Twitter follower of mine for sending that to me because, yes, I'm sure he remembered that we showed the original chart, which was fun also. But this is just kind of hysterical. So thank you.

Last week we talked about Chrome 85's release that day which would, among other things, fix a worrisome remote code exploit which existed in Chrome's WebGL rendering engine. But I said "among other things," so I wanted to take a couple of minutes to enumerate the other major changes that landed last week in 85. In this day of massively heavy web pages, as we know, speed is paramount. And as we've spoken of many times, Google has really focused, to their credit and to the Internet's benefit, on measures to increase the user's perceived browser speed just across the board, going so far as to pioneer new Internet communications standards like the QUIC protocol to enhance connection performance and the operation after a connection has been made. So big props and hats off to Google.

[Chrome] 85, which we've had now for a week, introduces a new compiler optimization technique known as PGO, standing for Profile Guided Optimization. It, in independent page-loading benchmarks, has shown a measurable and significant improvement in the browser's overall performance, that is Profile Guided Optimization. Chrome's Engineering Director Max Christoff said: "Because PGO uses real usage scenarios that match the workflows of Chrome users around the world, the most common tasks get prioritized and made faster." He added: "Our testing consistently shows pages loading up to 10% faster at the median, and even greater speed improvements when your CPU is tasked with running many tabs or programs."

So essentially PGO makes it possible for the browser's most performance-critical code to run faster. You know how browsers now have JIT, just-in-time compilers. Well, they're "just in time" meaning that uncompiled code is compiled as it's needed. But it turns out that, if you take a little more time to better optimize that code which is being compiled on the fly, the result, not surprisingly, is a much better page experience. And so essentially they're no longer treating all just-in-time compiled code equally. They are

using some compile time heuristics and some history of it in order to understand that it makes sense to take more time to better optimize this code which is being compiled just in time on the fly. So this ends up producing some great results.

There are two benchmarks which are often used. One is known as the "First Contentful Paint," which is the interval between the start of a page load and the browser's first display of any of the page's content. So of course that goes a long way toward describing the user's perception of how long they're waiting for the page. It's like, oh, good, here comes the page. Stuff is beginning to show. It measured a browser improvement of up to 3.5%, and there's something known as the speed meter browser benchmark, which clocked an 11.4 improvement, as well as an overall 7.3% improvement in the browser's responsiveness. So we just got that for free. And presumably, eventually, all Chrome instances, that is, all the other browsers that are Chromium-based will get that, as well.

And there's also an interesting new emerging new lossy image compression format known as AVIF. It is able to outperform WebP, JPEG, PNG, and GIF, and is intended to eventually replace them all. I loved this graphic of the AVIF imagine format support. This is something that anyone who tracks which features which browsers have will be used to seeing.

Among all of the different browsers - IE, Edge, Firefox, Chrome, Safari, Opera, iOS Safari, Opera Mini, and Android - and showing a stack of their various versions, there is exactly one green rectangle, which is Chrome 85. In other words, Chrome as of - actually it was as of, yeah, it was as of a week ago, now is the first browser and the only browser at the moment to offer support for this AVIF format. But again, as other browsers update their Chromium instances, we can expect more widespread support shortly.

This AVIF is AV1 Image File Format, which compresses using the AV1 codec and has been found to dramatically reduce image sizes without a significant loss of quality. So it's what you want. It's fewer bits describing a picture with equal crispness. In tests that were conducted by Netflix using AVIF images, they found that AVIF significantly reduces a file size while retaining high-level image detail. And not surprisingly, image size matters to Google because of course big images are going to take longer to load.

Google wrote: "Reduce bandwidth consumption to load pages faster and reduce overall data consumption." They said: "AVIF offers significant file size reduction for images compared with JPEG or WebP. Prior to full optimization, Netflix published results on their test, showing 50% savings versus JPEG across use cases, and going past 60% for images that were RGB." So anyway, it's compatible with high dynamic range images, and sites such as Netflix, YouTube, and Facebook have shown an interest in its use. So not surprisingly, Google decided to build in support for the file format; and I expect we'll see others, even non-Chromium browsers, following suit before long. For anyone who's interested in learning more about this, there's a link in the show notes, I put a link in to the Libre software site which has some very good background about it.

Over on the security side, about a year and a half ago, back in April of 2019, as part of their overall pro-HTTPS march, Google signaled that they would be tightening up their alerts and blocking of mixed content downloads. As we know, mixed content refers to any page asset which is fetched over, these days, HTTP from a base page that was fetched over HTTPS. For a long time, only so-called "active" mixed content was being blocked, and passive content such as images and MP4s and things, that is, for example, not an iframe, not JavaScript, certainly. Those are very active.

But passive things were perceived as being lots less worrisome and could just be allowed without comment. This was also likely a consequence of the fact that many sites were still continuing to pull some resources over HTTP. I'm sure that Google's Analytics demonstrated that, if 18 months ago they started just putting their foot down about

HTTP loading of passive content, it would break a lot of things. So today, or last week, with the release of 85, Chrome has started displaying a visual warning whenever mixed content audio, video, or images such as PNG, GIF, JPEG, and even MP4 videos are downloaded. Chrome is also now blocking other mixed content files that are considered unsafe because they could be abused to deliver malware, such as PDFs, DOCs, DOCX, and XLS and their like.

So assuming that this move last week doesn't result in some sort of a surprising end of the world as we know it, where many more things are being blocked and causing problems, the next Chrome, 86, is expected to block across the board all mixed content downloads, period, making no discrimination about what their nature is. So this is, essentially, this is Google saying, okay, we know that it was a pain to switch to HTTPS. But pretty much everybody is now. And boy, I'll tell you, anytime I do something that is not HTTPS, it's increasingly difficult. Like, you know, for example, Leo, you and I were talking about Syncthing. Well, if you're just talking to a web browser on your LAN that you're looking at across the room, there's arguably very little need to connect to it securely.

Leo: Right.

Steve: But, boy, does it complain, Chrome, if you try to talk to an HTTP thing. Or the same for our routers. Our routers are right here. Some of them have self-signed certs, but generally it's like, yeah, it's just 192.168.1.1 or something. And you could argue there's no point in it being HTTPS. But again, our browser's like, oh, I don't know. You might be up to no good here. It's like, oh, okay, fine. So anyway, the point is I'm sure that Google would say at this point it's past time that any resources on the 'Net should need to be HTTP. So we're just going to put our foot down because the only way this is going to get fixed, if we say no. And then if those end up being important, they'll be moved over to HTTPS. So we're sort of at that point right now where it's like, this is it.

In a couple other, or one other - one or two? Oh, yeah, two other features. Chrome 85 also allows you to create shortcuts for progressive web apps so that users will be able to quickly access commonly used tasks for which they use progressive web apps. You can hold your mouse over the PWA icon or right-click it, and you'll get a list of those that are enabled.

And also, back on the security and privacy side, we've talked often through the years about abuse of the user-agent header. Whenever our browser is making a query, one of the metadata tags in that query identifies a lot of features about the browser. And like so many features that were designed in the beginning as an aid to the web's operation, like cookies, for example, as an example of another one, the user agent string has been abused for tracking web browser users. It's an information-laden, relatively static thing that unfortunately has a lot of unique information.

The cool idea, and the reason it was originally created, was that the browser itself and its various add-ons could add their own designations and their version numbers to that string so that web servers receiving a query might become version number aware, and thereby able to perhaps work around known problems or lacks of features with specific versions, or probably back in the early days with specific browsers. It was like, oh, well, you know, this browser doesn't support this. So they would serve up a slightly different page, depending upon the profile of who was doing the asking.

Anyway, as I said, as a consequence of this, there's like some weird attributes that user agent strings have acquired over the years, like claiming that they are who they're not in order for them to receive content that they actually do know how to render.

So to combat this problem, Google had originally planned to enable the addition of a new feature called User-Agent Client Hints into last Tuesday's Chrome 85. But in another concession to the coronavirus, Google decided to drop this back into 2021, I think April, if I recall right. But, you know, next year. When it does happen, it will by default simplify the user-agent string so that it becomes much less juicy as another tracking signal. It is in Chrome 84, that is, previous to 85. It's just not enabled yet. But it can be switched on.

If you were to go to chrome://flags/#freeze-user-agent, that flag you can turn on which will essentially remove a lot of the extraneous information used for tracking today. So you can try it and see how it goes. Again, it's expected to be the default. And so at some point I'm sure Google will start experimenting to make sure that it doesn't break things that are mission critical.

In almost a, well, state-sponsored spy story, we have something that really happened. And I can tease this by quoting our friend Marcus Hutchins's Twitter reaction upon learning of it. Just to remind everyone, Marcus is the well-known security researcher and reformed cybercrime hacker. He actually reformed in his teenage years, but the FBI didn't forgive him for that. And of course, as we know, his future became uncertain when the FBI grabbed him in Las Vegas's McCarran Airport as he was departing or preparing to depart from the U.S. for his home in the U.K. following the annual Black Hat and Defcon conferences.

Well, last Thursday, reacting on Twitter to the news of this story which had just broken, Marcus quite correctly observed, he tweeted: "One of the benefits of cybercrime is criminals don't have to expose themselves to unnecessary risk by conducting business in person. Flying into U.S. jurisdiction to have malware manually installed on a company's network is absolutely insane."

Okay. So what was all that about? A 27-year-old Russian national by the name of Egor Igorevich Kriuchkov traveled to the U.S. and attempted to subvert and bribe an employee working at Tesla Corporation's massive Nevada-based Gigafactory. Egor ultimately agreed to pay the employee one million dollars to plant malware inside Tesla's internal network. The good news is the employee reported the offer to his employer, Tesla, and then worked with the FBI to build an airtight case and to set up a sting which included having him covertly record face-to-face meetings discussing this 27-year-old Russian's proposal.

In their complaint, which followed Egor's arrest and arraignment last Tuesday, the prosecutors wrote: "The purpose of the conspiracy was to recruit an employee of a company to surreptitiously transmit malware provided by the co-conspirators into the company's computer system, exfiltrate data from the company's network, and threaten to disclose the data online unless the company paid the co-conspirators' ransom demand."

The complaint said that the malware would be custom developed to propagate through the company's network. For it to work, the group said it needed the employee to provide information about the employer's network authorizations and network procedures. Kriuchkov said the malware would be transmitted either by inserting a USB drive into a company computer or clicking on an email attachment containing malware. Egor explained that the infecting computer would have to run continuously for six to eight hours for the malware to move fully through the network. To distract network personnel, a first stage of the malware would perform a denial of service attack, while a second stage performed the data exfiltration.

When the complaint was initially unsealed last Tuesday, the identities of all parties was still confidential, being identified only as "Company A" and "CHS1," which is their abbreviation for Confidential Human Source #1, that is, the employee. But last Thursday

Elon Musk confirmed that, yes indeed, it was his company that was the target of this whole operation. The charging document, which was filed in federal court in Nevada, detailed an extensive and determined attempt to infect Tesla's network. The defendant, again, 27-year-old Egor Igorevich Kriuchkov, allegedly traveled from Russia to Nevada and then met with the unnamed employee on multiple occasions. When Kriuchkov's initial $500,000 bid failed to clinch the deal, the defendant doubled the offer to $1 million.

According to the complaint, Kriuchkov wined and dined and boozed up the employee and, when discussing especially sensitive details, conducted conversations in cars. When FBI agents couldn't conduct physical surveillance in restaurants or bars, the employee recorded them. One meeting occurred on August 7th in a car Kriuchkov had rented. Referring to the employee again as CHS1, the prosecutors described that August 7th meeting as follows.

They said: "During this meeting, which the FBI had consensually recorded, Kriuchkov reiterated some of the details of the criminal activity previously proposed to CHS1. Kriuchkov described the malware attack as he did before, adding that the first part of the attack, a DDoS, would be successful for the 'group,' but the victim company's security officers would think the attack had failed. Kriuchkov" - and here's some news - "again listed prior companies this group had targeted. Kriuchkov stated each of these targeted companies had a person working at those companies who installed malware on behalf of the group. To ease CHS1's concerns about getting caught, Kriuchkov claimed the oldest project the group had worked on took place three and a half years ago, and the group's co-optee still worked for the company."

So in other words, this group has been active for three and a half years. If we're believing Kriuchkov, his assertions to this employee at Tesla, they have multiple times successfully bribed, presumably, and co-opted someone, an insider, getting them to work with them, to conspire, to insert malware into various U.S. companies to pull this off. And so of course we can imagine that now that Kriuchkov has been nabbed by the FBI, of great interest will be which are these companies and maybe who are the co-optees, as well.

Anyway, Kriuchkov also told the Tesla employee the group had technical staff who would ensure the malware could not be tracked back to the employee. In fact, Kriuchkov claimed the group could attribute the attack to another person at the victim company should there be someone in mind the employee wanted to "teach a lesson." During the meeting CHS1, the Tesla employee, expressed how concerned and stressed he had been over the request. He stated, if he were to agree to install the malware, he would need more money. Kriuchkov asked how much. The employee responded a million dollars.

Kriuchkov was sympathetic to the request and said he understood - remember, this is a recording that the FBI has and is part of this complaint - that he understood and would have to contact the group before agreeing to the request. Egor confided that the group was paying him half a million dollars for his participation in getting this employee to install the malware, and he was willing to give a significant portion of the payment, 300,000 to 450,000, to the employee to entice his involvement. The employee said he would need money upfront to ensure that Egor would not have him install the software and then not pay him. Again, Egor asked how much, and the Tesla employee responded $50,000. Egor said this was an acceptable amount, and a reasonable request, but he would have to work on this because he only had $10,000 with him due to U.S. Customs restrictions on the amount of money he could bring into the country.

Egor also questioned what would prevent the Tesla employee from taking the upfront money and then not following through on installing malware. The employee stated that he was sure Egor or the group would figure a way to apply leverage against the employee to ensure that he held up his end of the arrangement. They discussed the

timing of the next meeting, and Egor said he would return to Reno on or around August 17th of 2020.

So, yikes. This would have been a classic inside job; and it does serve as a reminder that, just as money motivates the bad guys, it's also, as we know, a time-honored motivator used to turn someone, anyone. And as I said, it's also a little bit sobering that if, again, if we believe what Egor was saying, this doesn't look like the first time this group has done this, and maybe the FBI will be able to get some information about other companies this has happened to. And you can imagine, if the FBI approached those companies, suddenly the company is wondering, okay, who was the co-conspirator in this? So a very, very interesting problem.

Leo: On with the show.

Steve: So we've got more problems with the EMV standard. We've talked a little bit about this over the years. EMV is the monetary transaction method based upon more than - and this is part of the problem - a 2,000-page specification.

Leo: Of course.

Steve: So if your security spec is 2,000 pages...

Leo: Who can read that? Who can follow that?

Steve: Exactly. And who can verify that the way it works is proper? It's the technical standard underlying the use of smart payment cards, payment terminals, and ATMs. EMV originally stood for Europay, Mastercard, and Visa, the three companies who created the standard. And of course that's part of the problem is, rather than making it an open academic cryptographer-laced process, it's more like the Wi-Fi Alliance that did it privately, and now they're sorry. So next May of 2021, researchers from ETH Zurich, the Swiss Federal Institute of Technology, whose work we've often covered in the past, they'll be delivering a paper at the IEEE Symposium on Security and Privacy titled "The EMV Standard: Break, Fix, and Verify."

The paper's been released in advance, presumably because this is really bad, and it cannot be readily, well, it can be readily patched and fixed, but keeping it a secret doesn't make any sense. So they're like saying, look, let's get this fixed. We're going to talk all about it because the coolness of their paper is the way they found the problems. It's important that the problems exist, as we'll see. But the mechanism of what they used is not going to lose any of its punch for waiting nearly a year. So it's not a matter of responsible disclosure.

It's also not the end of the world since the attacks so far designed still require physical proximity to a Visa card. But the attack described does completely bypass the system's built-in security measures of the PIN which is designed to prevent abuse of a stolen Visa card. It also bypasses completely any payment transaction limits which are designed to limit the damage in the case of abuse. So hopefully this has got the attention of the relevant parties.

The paper's abstract which describes their research reads: "EMV is the international protocol standard for smartcard payment used in over nine billion cards worldwide.

Despite the standard's advertised security, various issues have been previously uncovered, deriving from logical flaws that are hard to spot in EMV's lengthy and complex specification, running over 2,000 pages. We formalize," they wrote, "a comprehensive symbolic model of EMV in Tamarin, a state-of-the-art protocol verifier. Our model is the first that supports a fine-grained analysis of all relevant security guarantees that EMV is intended to offer. We use our model to automatically identify flaws that lead to two critical attacks, one that defrauds the cardholder and another that defrauds the merchant.

"First, criminals can use a victim's Visa contactless card for high-value purchases without knowledge of the card's PIN. We built a proof-of-concept Android app and successfully demonstrated this attack on real-world payment terminals." In other words, once this model found flaws, they then said, oh, we know how to do that. And so they wrote some Android apps - that actually takes two apps and two phones, I'll explain all that in a minute - which pulled off the heist. It works.

"Second," they said, "criminals can trick the terminal into accepting an unauthentic offline transaction, which the issuing bank should later decline, after the criminal has walked away with the goods. This attack is possible for implementations following the standard, although we did not test it on actual terminals for ethical reasons." In other words, it would have defrauded the merchant, they said, though it seems like that could have been handled by engaging a willing and interested merchant and, like, giving back the goods after the bank declined the transaction.

In any event, they said: "Finally, we propose and verify improvements to the standard that prevent these attacks, as well as any other attacks that violate the considered security properties. The proposed improvements can be easily implemented in the terminals and do not affect the cards in circulation." So that's important. We cannot update nine billion existing Visa cards, nor do we have to. The terminals can be fixed, and hopefully they're connected to a network, and they can be updated online, over the network. So it looks like this can be remediated globally without much trouble. Anyway, so this EMV standard, as I noted, was developed by another closed group, and this is what you get when that happens.

To establish a bit of background, the researchers wrote: "EMV, named after its founders Europay, Mastercard, and Visa, is the worldwide standard for smartcard payment, developed in the mid-1990s." And truthfully, given that it's, what, 25 years old, I guess I cut it a little bit of slack. They certainly did know how to do things then as well as we do today. And what is so cool is that we're beginning to see the emergence, we've covered some already recently, of applying not quite AI, but automated protocol verification to bring real guarantees of robustness to protocols.

Anyway, they said: "As of December 2019, more than 80% of all card-present transactions globally use EMV, reaching up to 98% in many European countries." In other words, 80% as of December 2019. So basically that's what transactions are using is this EMV protocol. "Banks have a strong incentive to adopt EMV due to the liability shift, which relieves banks using the standard from any liability from payment disputes. If the disputed transaction was authorized by a PIN, then the consumer is held liable. If a paper signature was used instead, then the merchant is charged." But in neither case is the bank responsible.

"So besides the liability shift, EMV's global acceptance is also attributed to its advertised security. However, EMV's security has been challenged numerous times. Man-in-the-middle attacks, card cloning, downgrade attacks, relay attacks, and card skimming are all examples of successful exploits of the standard's shortcomings." So in other words, yes, it's 25 years old, and it is showing its age because it was never rigorously developed, and it was developed in a closed setting.

They said: "The MITM [man-in-the-middle] attack is believed to have been used by criminals in 2010 and 2011 in France and Belgium to carry out fraudulent transactions totaling 600,000 euros. The underlying flaw of the attack is that the card's response to the terminal's offline PIN verification request is not authenticated. Some of the security issues identified result from flawed implementations of the standard. Others stem from logical flaws whose repairs would require changes to the entire EMV infrastructure." In other words, meaning it's too late to fix them now.

They said: "Identifying such flaws is far from trivial due to the complexity of EMV's execution flow, which is highly flexible in terms of card authentication modes" - in other words, this suffers from the kitchen sink problem of security. Also they said "cardholder verification methods, and online/offline authorizations." Again, they just tried to do everything with this. "This raises the question of how we can systematically explore all possible flows and improve the standard to avoid another 20 years of attacks." So that's what they did.

They explain: "In this paper we focus on weakness and improvements to the EMV protocol design. We present a formal, comprehensive model for the symbolic analysis of EMV's security. Our model is written in Tamarin, a state-of-the-art verification tool that has been used to study numerous real-world protocols, including TLS 1.3 and 5G authentication. Tamarin supports protocol verification in the presence of powerful adversaries and many concurrent protocol sessions without bounds. Our model supports the analysis of all properties that must hold in any EMV transaction. An informal description of the three most relevant properties are: Bank accepts terminal-accepted transactions: No transaction accepted by the terminal can be declined by the bank. Authentication to the terminal: All transactions accepted by the terminal are authenticated by the card and, if authorized online, by the bank. And third, authentication to the bank: All transactions accepted by the bank are authenticated by the card and the terminal."

They said: "Our model faithfully considers the three roles present in an EMV session: the bank, the terminal, and the card. Previous symbolic models merge the terminal and the bank into a single agent. This merging incorrectly entails that the terminal can verify all card-produced cryptographic proofs that the bank can. This is incorrect, as the card and bank share a symmetric key that is only known to them. Using our model," they said, "we identify a critical violation of authentication properties by the Visa contactless protocol. Specifically, the cardholder verification method used in a transaction, if any, is neither authenticated nor cryptographically protected against modification.

"We developed a proof-of-concept Android app that exploits this to bypass PIN verification by mounting a man-in-the-middle attack that instructs the terminal that PIN verification is not required because the cardholder verification was performed on the consumer's device. This enables criminals to use any stolen Visa card to pay for expensive goods without the card's PIN. In other words, the PIN is useless in Visa contactless transactions."

So in practice they used a pair of standard Android smartphones, each running a custom app that they wrote. The smartphones do not need to be hacked. No root privileges or anything fancy is required. Just generic custom Android apps which successfully ran on phones. In this case they used them both on Pixel phones from Google and also handsets from Huawei. The two phones are linked by WiFi so they can talk to each other. One phone uses its NFC radio to interact with the stolen card, thus pretending to be and emulating the point-of-sale terminal which the card thinks it's talking to, while the other phone uses its NFC radio to emulate the payment card to the authentic point-of-sale terminal. Thus the point-of-sale terminal thinks it's talking to the card. It's instead talking to the Android phone's NFC.

So you can see what this does is each of the devices, the card and the terminal, normally have an NFC link to each other. Instead, we separate them with an Android phone-based link, allowing the protocol to be modified on the fly. And the point is this system is unfortunately weak enough that the protocol can be tweaked as it moves between the two Android phones to convince the card that it's talking to the point-of-sale terminal, and to convince the point-of-sale terminal that the PIN has been managed between the user and the card. Therefore the terminal need not ask for a PIN. And there's another protocol flaw that allows a limit which is normally imposed on PIN-based transactions, exactly for this purpose, to simply be bypassed. So an unlimited size transaction can be performed against an NFC-linked EMV terminal used in 98% of transactions in the EU and 80 globally, against any Visa contactless payment card, without needing the PIN from the owner.

As I said, this is an important piece of work. They took a 2,000-page spec which is mind-boggling complex, reduced it to a symbology, which then allowed them to code this into their research app that was able to analyze the resulting protocol, spot the flaws, and from that they developed a proof of concept which successfully works. And hopefully it will be possible to - they indicated a fix was possible in the point-of-sale terminal, which with any luck will be going out to all of these point-of-sale terminals globally before much longer.

So anyway, very, very cool piece of work from these researchers in Switzerland. And really sort of a demonstration of the way we're seeing high-level analysis of security protocols now being performed. It's not just ad hoc analysis, especially for things that are as complex as this. You develop a means for turning a computer loose on it and allowing it to highlight problems and find them for you. And then you go manually look to see what's going on.

On the flipside, we've seen something sort of like that with directed fuzzing, where we use a computer to throw just junk at an API. And if it manages to crash the system, then you bring the humans in to figure out what it was that did the crash, reproduce it, and then see if it's possible to instrument it. So very cool.

I have some miscellany. Some closing-the-loop stuff, but some miscellany. Back in the early days of the COVID-19 pandemic I shared a few YouTube videos that seemed important. Thanks to the use of my grc.sc redirect links, I'm able to get some sense for our listeners' interest in various topics. 5,963 of our listeners visited the first of those COVID links I shared. That was that excellent Ars Technica guide in the very early days that provided some information, very useful. 6,376 visited the second, which was the "coronavirus explained" video. And 5,942 visited the third, which was that whiteboard, that medical school class grade whiteboard on COVID-19. And those numbers place COVID-19 or interest in it at the top of the historical counts of the links visited.

So today I have a fourth, which is frankly an astonishing video produced by a hardworking medical doctor researcher explaining the operation of the three primary technologies employed for testing for the presence of the COVID-19 virus or antibodies. The link is grc.sc/covid4, C-O-V-I-D-4. Again, grc.sc/covid4. I believe that - aside from being seriously educated about the operation of these tests, frankly in way too much detail, so don't worry about understanding it all the first time you watch it. Just sort of let it wash over you. I was really impressed. Anyone watching this will come away with a deep appreciation for the complexity of testing, a sober sense for how much medical science has successfully reverse-engineered our genetics. It's just really cool.

And perhaps some better sense for why it appears that so far testing for COVID-19 has been seriously botched. It is a truly delicate, error-prone, and error-fraught process. You'll get many clues about that from watching this video. It's about a 43-minute-long video. And testing just could not succeed in any environment of corner-cutting or rushing

to get a result. But anyway, don't take my word for it. If you watch the video, you'll understand. So grc.sc/covid4. It's probably too much for some people who aren't interested in details. But it's really, really compelling. And again, it's clear that our listeners had an interest in it, and so I wanted to extend this. This just came to my attention. Somebody posted it over in the grc.health newsgroup. And so it's just - it's excellent.

I received a DM from a listener saying: "Hi, Steve. I was wondering if storing password manager database in Google Drive or Dropbox is safe." And so I suggested that the best thing to do would be to first use the 7-Zip archiver with a strong password. We talked about it recently. Well, first of all, it'll scrunch the password database way down. And once it's encrypted with a strong password, it won't matter who obtains it. This is the old TNO approach of Trust No One, where you encrypt before it leaves your system.

And we talked about 7-Zip recently. I looked deep into it again. They did all the encryption right. They use a strong password-based key derivation function based on SHA-256 to generate the encryption key. And at that point, once you've done that, it's probably the password manager that's keeping all of those private is the weakest link because they're in active use. If you create a really well-encrypted archive with a strong password, your stuff is safe. And then by all means, you can put it anywhere, pretty much.

Alan Kopp tweeted: "A little puzzled by your recommending Syncthing last week. Are you really okay with it doing its connecting over the Internet without a VPN?" And Leo, this links back to what you and I were talking about before the podcast. And I replied to him. I said: "Yes, I'm okay with Syncthing because of the way it's designed. All endpoints can, and typically do, sit safely behind a NAT router, so there's no port open to be scanned. A set of rendezvous servers out on the public Internet receive outbound pings from the Syncthing endpoints, and all communication is over TLS v1.2 with the rendezvous server's certificates pinned by the clients."

So the privacy of the conversations is as strong as a VPN's. If a bad guy were to compromise one of the rendezvous servers, they could send their ID, their endpoint ID, to an endpoint and ask to be connected. But that's the extent of the threat. As anyone using Syncthing knows, you get a notice saying this endpoint wants to connect to you and share stuff. And you have to permit it in order for anything to happen. So you would just see this bogus request and say no. I mean, like in the worst case, that's all that would happen.

And Syncthing clients are able to use public STUN servers, S-T-U-N servers, to knit together direct NAT-to-NAT connections between peers that are both located behind NAT routers. We discussed the STUN and TURN protocols to support NAT many years ago. And if direct NAT traversal fails, relay servers are also available for inter-client relaying. But even then, since all data is truly end-to-end encrypted between client endpoints with certificates, there's no exposure of the relayed data to the relay server. So yes. As we said, I mean, I'm using it to transport mission-critical stuff. Leo, I know you are. And I do so without question. These guys - oh, and all of the protocol is open. They have beautifully documented four different specific protocols. Like the block transfer protocol is laid out. It's explained. It's like it's RFC-like in its thoroughness. So I'm just very impressed with Syncthing. And I use it without concern.

Leo: Oh, I'm so happy because I use it religiously for everything. Everything.

Steve: Yeah. It's just a win.

Leo: And I think I'm going to do that - kind of this ties your previous question together, do some pre-Internet encryption on a Syncthing folder because there's files I want to get on all my computers, but I'm just nervous about having them visible in any way. If I use 7-Zip, for instance, to password encrypt them, and then sync that blob, that would be even a bit safer, wouldn't it. Things like my dotfiles, my PGP passwords, things like that. Not passwords [crosstalk].

Steve: Yeah, I would say, you know, if they're really critical, and they don't need to have like automated access, if for example you want to have access to them, but you'd be unpacking them and sticking them in the root directory of a new Linux build or something, then it would make sense.

Leo: Yeah, exactly, because that's what I do.

Steve: If it doesn't have to be dynamically modified.

Leo: Yeah, that's exactly what I do, yeah. I keep standardized dotfiles. Any time I do a new setup I want to have that there. So that's perfect. Good. Good.

Steve: Yeah. So...

Leo: I'm glad you mentioned Syncthing. I love it. I needed your [crosstalk] approval.

Steve: Yeah, it's good. And just the ability to build an ad hoc peer-to-peer network, I mean, it's just - that's very cool.

Leo: Yeah, with no third party.

Steve: So Jon tweeted, someone named Jon tweeted via DM. He says: "Do you have a recommendation on a Zinc supplement?" He said: "My doctor suggested I add a zinc supplement to my vitamin regimen." He says: "Thanks for all the health insights you've shared on Security Now! over the years." And so I do have something useful, probably to a lot of our listeners. There's only one, or at least one very clear way to choose which one from among many. And I'll just give a little brief bit of what I've learned about this since I crossed this bridge 15 years ago. The most important thing to appreciate is that not everything we swallow is absorbed the way we intend across our intestinal lining to make it into our bloodstream. And things that do not naturally occur in food, like a mineral supplement, are often not readily absorbed because we're trying to fool Mother Nature. As it turns out, in this instance she can be fooled.

I looked at this 15 years ago, after I read several books about magnesium. I became convinced, as I remain today, that I wanted to get as much magnesium into me as I could for the rest of my life. For reasons maybe I'll discuss someday, it's really crucial. And that's what I've been doing ever since. But it turns out that's easier said than done. Since this is not a health and nutrition podcast, I won't spend the hours talking about it that I easily could because it fascinates me. Maybe someday.

So here's the bottom line. Most mineral supplements are packaged as a simple salt of the mineral - zinc citrate, zinc picolinate, zinc gluconate, for example, in the case of zinc. The problem with all of these simple salts of zinc, or magnesium, for that matter, is that those compounds quickly disassociate in the acidic low pH environment of our stomach. After that, we have atoms of the mineral floating around loose and not being well absorbed by the lining of our upper small intestine. One company named Albion Minerals, A-L-B-I-O-N, Albion Minerals, figured out how to solve this problem. They produce a large range of raw material, both for animal (veterinary) and human consumption. They turn the raw mineral into a dipeptide, which is the mineral bound to two amino acids. Glycine, which is the smallest of the amino acids, is the preferred choice.

So there exists a substance known as zinc bisglycinate, consisting of an atom of zinc held in by two glycine molecules. What's special about this complex is that our digestive system sees it as an organic molecule rather than a mineral. And unlike any of the salts, it's resistant to disassociation, and it remains intact as a consequence in our stomach's acidic low pH environment. That allows it to progress as a whole into our intestines where our intestinal lumen's active transport disassembles it and moves it into our bloodstream. The result is that a far greater percentage of what we swallow makes it into us.

Albion is not a retailer, so you'll see no brands, you won't see any supplements by them. They exclusively manufacture the bulk supplement raw material. So you'll find their name and their trademarks, Albion and TRAACS, T-R-A-A-C-S, which is some abbreviation for something, some means of measuring it, on the supplements. Doctor's Best uses them, Healthy Origin uses them, and a number of other supplement makers do. So for what it's worth, when shopping for mineral supplements, look for the words Albion or TRAACS. I jumped onto Amazon, googled Albion zinc, and found an inexpensive zinc supplement which I responded to Jon. And it was zinc bisglycinate, available from a [crosstalk].

Leo: And what does the zinc do, just out of curiosity? I'll take as given I want it, but...

Steve: Yes. Zinc is a useful supplement. It provides immune system support, for one thing.

Leo: Oh, that's probably a good thing to have, yeah, nowadays.

Steve: Yeah. So it's a good thing to have these days.

Leo: I take the magnesium you recommended that's also TRAACS.

Steve: Yes.

Leo: As you mentioned, yeah.

Steve: Exactly. I consume it in great quantities.

Leo: You eat a lot more than I do. I do two tablets in the morning and at night. What do you do, like five twice a day, something like that?

Steve: I do, yes. I do five twice a day. But I'm glad you're taking two, Leo. It's good.

Leo: Better than nothing. It can affect your digestive system a little bit, so you want to kind of work your way up.

Steve: Well, exactly. That's how you determine where your limit is.

Leo: Yeah.

Steve: You'll know when you've taken too much.

Leo: You'll know, yeah, yeah. Let's just put it this way. It's the same ingredient in Milk of Magnesia. Give you some idea. Okay.

Steve: One of my high school buddies, two of them became MDs. One is a lifelong ER doc. And in fact he lives up in Healdsburg, where he's only recently been allowed to return to his wonderful ranch on a vineyard as a result of the fires up there recently.

Leo: It's been so bad, yeah.

Steve: Anyway, but he's a lifelong ER doc, and he calls magnesium the "miracle mineral."

Leo: Really. Wow.

Steve: So apparently it's been of great use for him in helping people who are in emergent trouble.

Leo: I take it on faith. You tell me to take it, I take it.

Steve: I'm glad you do.

Leo: Yes.

Steve: So anyway, lastly, I got a tweet from LOLPANDA, who said, "Hi, Steve. I'm sorry to tweet here" - I don't know why he's apologizing. That's what Twitter's for. But, he says: "I'm so excited to say hello and thank you!! Once again SpinRite helped me by saving hours of work. SpinRite isn't only a matter of zero and one, it's magic. Trillion thanks."

Leo: Nice.

Steve: And by way of update I can finally report that the development of the AHCI driver has reached a new and very welcome stage. Test release 29a of the AHCI benchmark was posted to the group last week, and it has been rather exhaustively tested over the weekend. For the first time ever, there has not been a single problem that anyone has found running it on any of their various PCs. So it appears to be ready for its much larger and broader testing debut. Once this podcast is conducted, or concluded, conducted and concluded today, I'll be integrating the earlier IDE hardware driver work into the new AHCI driver to produce a single benchmark that should run at maximum possible speed on any drive and be able to benchmark any drive's read performance on any PC, and on older Macs that offer the Boot Camp booting option.

We've got a bunch of Mac users who have been testing this along the way. And in fact I bought a Mac, an older MacBook Pro, because there was a weird power management behavior that I was unable to untangle remotely, and I did after I got one. So just for the record, so people don't go off looking for it, I don't have anything yet. There's no download link.

I need to integrate things. I'll end up building an app which, like the InitDisk, will prepare a bootable USB thumb drive that you can use to boot any computer you want to and benchmarks its hard drives at absolutely their maximum link and throughput speed. Probably surprise you with some of the things that you find based on some of the things that we have found. And in the process you'll be further testing this driver, and I'll have a web forum set up in order to solicit feedback from users. And that's like the final stage before this moves into SpinRite, and we get 6.1. So we're getting there.

Leo: Steve, what did I do last summer? Because I don't remember. It's all a blur.

Steve: I don't think any of us - I think a lot of us did very little this last summer.

Leo: Yeah, that's right.

Steve: There are two pieces of research, one conducted eight years ago in 2012, and a similar very closely related research which was presented just last month during the USENIX 16th Symposium on Usable Privacy and Security. The early paper from 2012 was titled, interestingly, "Why Johnny Can't Browse in Peace." Okay?

Leo: Okay.

Steve: On the uniqueness of web browsing history patterns. It explains its purpose and its findings as follows. They wrote: "We present the results of the first large-scale study of the uniqueness of web browsing histories, gathered from a total of [a lot] 368,284 Internet users who visited a history detection demonstration website." And remember we've previously talked about how this can be done. Since our browsers will color previously visited URL links differently from ones it has not seen before, it's possible for a sneaky website and server to remotely probe our browser's site-visiting history by placing test URLs into the DOM (Document Object Model) and then using the Canvas API to read out the rendered color of those links.

Again, not anything any of the designers of these APIs ever intended to have happen. But as we keep seeing, where there's a will, there's a way. And the more sophisticated and complex we make our browsers, the more they become little like Turing complete systems that can do all kinds of unexpected and unintended things.

Anyway, in their abstract they wrote: "Our results show that for a" - okay, so just backing up. 368,000-plus Internet users visited this history detection demo website, which sucked, essentially and effectively, sucked the browsing history out of their web browser, which is not supposed to be available to sites you visit; right? That's none of their business. But again, it can be done. They said: "Our results show that for a majority of users (69%), the browsing history is unique, and that users for whom we could detect at least four visited websites were uniquely identified by their histories in 97% of cases." In other words, where we steer our browsers is surprisingly unique. And I know my own browser use. Yeah, I go to a few sites, like DigiKey and DigiCert, that lots of other people are not going to be going to directly. So I can see that.

They said: "We observe a significant rate of stability in browser history fingerprints. For repeat visitors, 38% of fingerprints are identical over time, and differing ones were correlated with original history content, indicating static browsing preferences. We report a striking result that it is enough to test for a small number of pages in order to both enumerate users' interests and perform an efficient and unique behavioral fingerprint. We show that testing 50 web pages is enough to fingerprint 42% of users in our database, increasing to 70% with 500 web page tests.

"Finally, we show that indirect history data, such as information about categories of visited websites, can also be effective in fingerprinting users" - sort of like taking a meta view of websites, classifying similar websites, and then using that as the fingerprint. That's also effective to fingerprint users - "and that similar fingerprinting can be performed by common script providers." Again, "...similar fingerprinting can be performed by common script providers such as Google or Facebook."

Leo: Hmmm.

Steve: Okay. So in other words - uh-huh, uh-huh. It's not just cookies, and it's not just obvious things. In other words, this introduces another entire category of tracking signal and/or tracking reacquisition in the event of third-party cookie blocking or deliberate cookie deletion. Our browser histories turn out to serve as a surprisingly powerful disambiguator.

Leo: That makes sense, yeah.

Steve: It does make sense. And as I mentioned, that research was followed up on and recently updated by a three-person team at Mozilla. Their USENIX paper from a couple weeks ago was titled "Replication: Why We Still Can't Browse in Peace: On the Uniqueness and Reidentifiability of Web Browser Histories." And they explain their work and their findings a little more briefly as follows. They said: "We examine the threat to individuals' privacy based on the feasibility of reidentifying users through distinctive profiles of their browsing history visible to websites and third parties." Again, that's the key. Visible to websites and third parties.

They said: "This work replicates and extends the 2012 paper 'Why Johnny Can't Browse in Peace: On the Uniqueness of Web Browsing History Patterns.' The original work demonstrated that browsing profiles are highly distinctive and stable. We reproduce

those results and extend the original work to detail the privacy risk posed by the aggregation of browsing histories. Our dataset consists of two weeks of browsing data from about 52,000 Firefox user volunteers. Our work replicates the original paper's core findings by identifying 48,919" - now remember, out of 52,000, 48,000, almost 49,000, 48,919 - "distinct browsing profiles, of which 99% are unique. High uniqueness holds even when histories are truncated to just 100 top sites."

Leo: Wow.

Steve: Uh-huh. We then find...

Leo: This is really not surprising, really.

Steve: Yeah. I agree, Leo. I mean...

Leo: What's surprising is that they can read it.

Steve: Yes.

Leo: That's what's annoying.

Steve: Uh-huh, exactly.

Leo: Of course, if you have it, you can make - it's probably very unique.

Steve: Yup.

Leo: 100%.

Steve: "We then find that for users who visited 50 or more distinct domains in the two-week data collection period, about 50% can be reidentified using the top 10,000 sites. Reidentifiability rose to over 80% for users that browsed 150 or more distinct domains. Finally, we observe numerous" - so basically what that's saying is that, if in a short period of time not many total domains were visited, that is, some people just didn't roam broadly, then there just isn't enough virtual...

Leo: Right, uniqueness, yeah.

Steve: ...uniqueness, yes, in order to identify them. But if a person tends to roam around a lot more to, for example, visit 150 different distinct domains, then they become much more, 80% reidentifiable. And they said: "Finally, we observe numerous third parties pervasive enough to gather web histories sufficient to leverage browsing history as an identifier." So in other words...

Leo: Remember when we talked about how Google was doing the work of god by taking third-party tracking cookies out of Chrome finally?

Steve: Uh-huh.

Leo: But we surmised at that time, that's only because they have a better way of fingerprinting you. And they don't want the amateurs doing it. Let us fingerprint users. And of course this proves it.

Steve: Yes. So exactly. The overt privacy problem of cookies that everyone looks at turns out to be like, okay, fine, block them. We don't care. And you'll notice that DNT didn't go anywhere because it didn't specify how you track, just please don't. And so it's like, no, we don't really want to do that. We don't want people to - we don't want to have to honor someone saying don't track me because we'll just say, okay, wouldn't you like those cookies removed? Yeah, we'll scrape those off for you.

So it is discouraging. There are two ways that our histories can be obtained. They can be obtained by sucking them out of our browser, for which the technology exists. All of these different entities - oh, Leo? I discovered a creepy site called DoubleVerify.com. If you go to DoubleVerify.com and just sort of - this is one of the people, one of those that we sometimes see. They tend to stay in the shadows. They don't like to be seen. But I picked up on them when I was doing this research about what sites are there doing this behind our backs? It is a super slick-looking site. But it's a little bit creepy when you sort of understand that these are the people who are putting little script snippets in ads and on websites.

Leo: But they're in the business of trust, Steve.

Steve: Oh, oh. I missed that, Leo. I didn't see where it said that. Uh-huh.

Leo: How could they be bad? How could they be bad? By the way, this stupid Accept Cookies popup, of course. Now we know that's meaningless. Yeah, sure. Whatever. Go ahead, put cookies on there. That'll stop them.

Steve: Yeah, that'll keep them pacified.

Leo: Yeah. Oh, man.

Steve: Yeah.

Leo: Yeah. There's a lot of this in the world. They call these "tracking pixels." And there's all sorts of ways to do this, even though - in fact, lots of them are not pixels anymore. That's just kind of that's how they used to do it.

Steve: Right. They drop little bits of script now, and they're able to do much more.

Leo: Much more, yeah.

Steve: It's very much like the Google Analytics that it's so useful for the website because Google tells you all kinds of cool stuff. But it's also Google running their own script on every single one of those pages out in the world. Yeah.

Leo: Really interesting, yeah.

Steve: Yeah. So that and an ISP monitoring our DNS queries. That's the other way you obtain browser histories is by looking at the DNS lookups that all of your clients are using. And so another sort of little nudge towards using encrypted DNS in the future, which looks like it's where we're all headed. So anyway, I thought that was really, really interesting. I just wanted to put it on all of our listeners' radar that, yes, maybe it's worth flushing our browser histories. It's sad, too, because I really like the fact that my links are, you know, when I do a search for something that's related to something I've searched for before, I see some that are purple, and I go, oh, I've already been in there. No need to go again. And, oh, Google's a little spooky because it'll say, oh, you were there three days ago. It's like, okay. Yeah, well, I am being - I'm being tracked.

Leo: Well, you know, also I'm a little sympathetic...

Steve: It's free. Everything's free, Leo.

Leo: Yeah, I'm a little sympathetic. We do a little bit of that ourselves, to be in full disclosure, because podcasts you really can't do tracking pixels, as you might imagine. But that's one - remember we were talking earlier about redirects. One of the redirects goes through a company called, I can't remember the name. Want to say Chartbeat. I can't remember. But it goes through a company that does an interesting thing, and I think this is - our advertisers say you've got to do something. So we don't do it for everybody. They have to pay for it. We don't. They don't get any information, which is the good news.

But what happens is this company gets, effectively, our logs. They get the redirects through them. So they track the IP addresses. They store those of all the downloads. But they don't give them to anybody. We're very careful to make sure. This is not public information. It's the same stuff that we've been using, we send to Podtrac, the same exact stuff. And then if a company wants to contract with them for an ad campaign, let's say LastPass says we want to see if this drove any traffic, of course we always use those URLs, and we hope people will use them. But they're not - companies say, well, we don't know if we trust them. Most companies, most of the URLs tend to be TWiT, even though like if I have the Security Now!, people are still going to use TWiT instead as the offer code.

Steve: Right, right.

Leo: So everybody says, well, TWiT works, even if they've never been on TWiT. They say, well, that's the one that works the best because everybody uses that. So you lose some credit. So a company, if they want to do more informational tracking,

will then put the tracking pixel from this company, Chartable I think, on their site. And then that redirects IP addresses of people visiting their site or their landing page to Chartable. And Chartable does a matchup.

Steve: Ah, right.

Leo: And without sending IP address information to the advertiser, or disclosing it, they say 82% of the people that visited your site in this three-week period also downloaded Security Now!.

Steve: Nice.

Leo: So I consider that relatively benign. We have to do something because advertisers are really, nowadays, I mean, look, we're competing against Facebook and Google, who tell them everything; right? They know everything about you. We know nothing about you except for the fact that you've downloaded that show. So I think this is a relatively benign way, without disclosing any information about you to any third party, matching those IP addresses up gives them some knowledge about how successful their campaign was. And at this point [crosstalk].

Steve: Well, and also bouncing through multiple redirects tends to be an anonymizing thing anyway, I mean, so...

Leo: Right, right, yeah. So that's, you know, we bounce it through I think two redirects right now. One is our own for counting because we charge people based on how many people listen, so we need to count that. It's a cost per thousand is how we work. And then we do this additional Chartable redirect so that people can - I think it's Chartable. I hope I'm not saying the wrong name. We went through three different companies to find one that really was privacy forward and would do this respectably. And we think we've found one, and I can't remember who it was. But gosh, I'm probably saying the wrong name. Anyway, this company matches it up without giving up any information of yours to anybody else. And I think that's benign; right?

Steve: Yeah.

Leo: Does it sound like...

Steve: I think that's as benign as it could be.

Leo: Yeah. The beauty of RSS, the reason Spotify's buying podcasts is because then you listen in their app. That's why they make them exclusive. And they know everything about you. We don't know anything about you. We just know you downloaded the show. And all we know is the IP address you used when you downloaded it. That's it. And frankly, that's all we'd ever want to know. And I don't want to know that much, but it's inevitable. That's how you count. You have to count unique IP addresses or you wouldn't know the downloads; right? Because very

frequently, for instance, you use Apple's podcast client, it'll open 10, it'll open 10 different parts of the podcast at once. So is that 10 downloads? No, that's one. And so you can't just count hits. So anyway, I'm sorry. I didn't meant to - in full disclosure we do something like that.

Steve: Yeah.

Leo: But I think we do it in a way that's as benign as possible. Anyway, if you hate that, then use VPN, ExpressVPN. No one will know. No one will know. It's completely up to you.

Steve, once again you have illuminated and explicated and pontificated. You've done all the -ateds. And that means we're done. You can go to GRC.com, where he does not track you. Do you even keep logs, web logs? Probably not; right?

Steve: I don't have logs. I don't have logging on, no.

Leo: People think that their IP address is a super secret thing. It's every website you go to gets it. Obviously, they couldn't open a conversation. And most web server software by default logs. You would have to explicitly go to IIS and say, I don't want to know. Don't keep track. It's just filling up my hard drive with useless information.

So go there. It's safe. GRC.com. You'll find lots of stuff there including SpinRite, the world's best hard drive maintenance and recovery utility. If you buy 6.0 now, 6.1's coming soon, and you'll be there. Free upgrade. He also does a lot of great free stuff, including all this vitamin stuff is there, the health stuff, if you want to see that. And the podcast, 16Kb and 64Kb audio. Plus, and it's unique to Steve's site, transcripts, really nicely done by a human transcriber, Elaine Farris, who does a great job with these. You'll get that, and she's doing it right now. Stop typing. Elaine, stop typing.

Steve: She also corrects my mistakes, which is really handy. I get the years wrong sometimes.

Leo: Oh, that's handy. And does she leave out the ums and uhs? Probably not.

Steve: I don't remember. We discussed that a long time ago. I hope so.

Leo: You see some of those in there every once in a while. That's fine. Hey, we want a record of the show.

Steve: It humanizes it.

Leo: It's also really great for search because you can do a text search on it and find that part of the show very easily. A lot of people like to read along while they listen. That's all at GRC.com. We have 64Kb and video at our site, TWiT.tv/sn. You can also subscribe in your favorite podcast at a client. You can use Spotify if you want, but

it's an RSS feed, so they don't - you can use anything you want, anything that'll take RSS feeds, and you'll automatically get it the minute it's available. Subscription's the best because then you don't miss a single episode.

If you want to watch us do it live, it's Tuesdays, 1:30 Pacific, 4:30 Eastern, 20:30 UTC. The live audio and video streams are at TWiT.tv/live. Those go all day and all night. So you can always hear what's going on in the studio. About an hour from now it'll be All About Android coming up. TWiT.tv/live. Chatroom, if you're watching or listening live, is irc.twit.tv. They're doing the same.

Offline we have a forum, just like Steve does. That's our TWiT Community at www.twit.community. So that's for people who are listening asynchronously.

Steve, have a wonderful week, and I'll see you next time.

Steve: Thank you, my friend. Right-o. Bye.