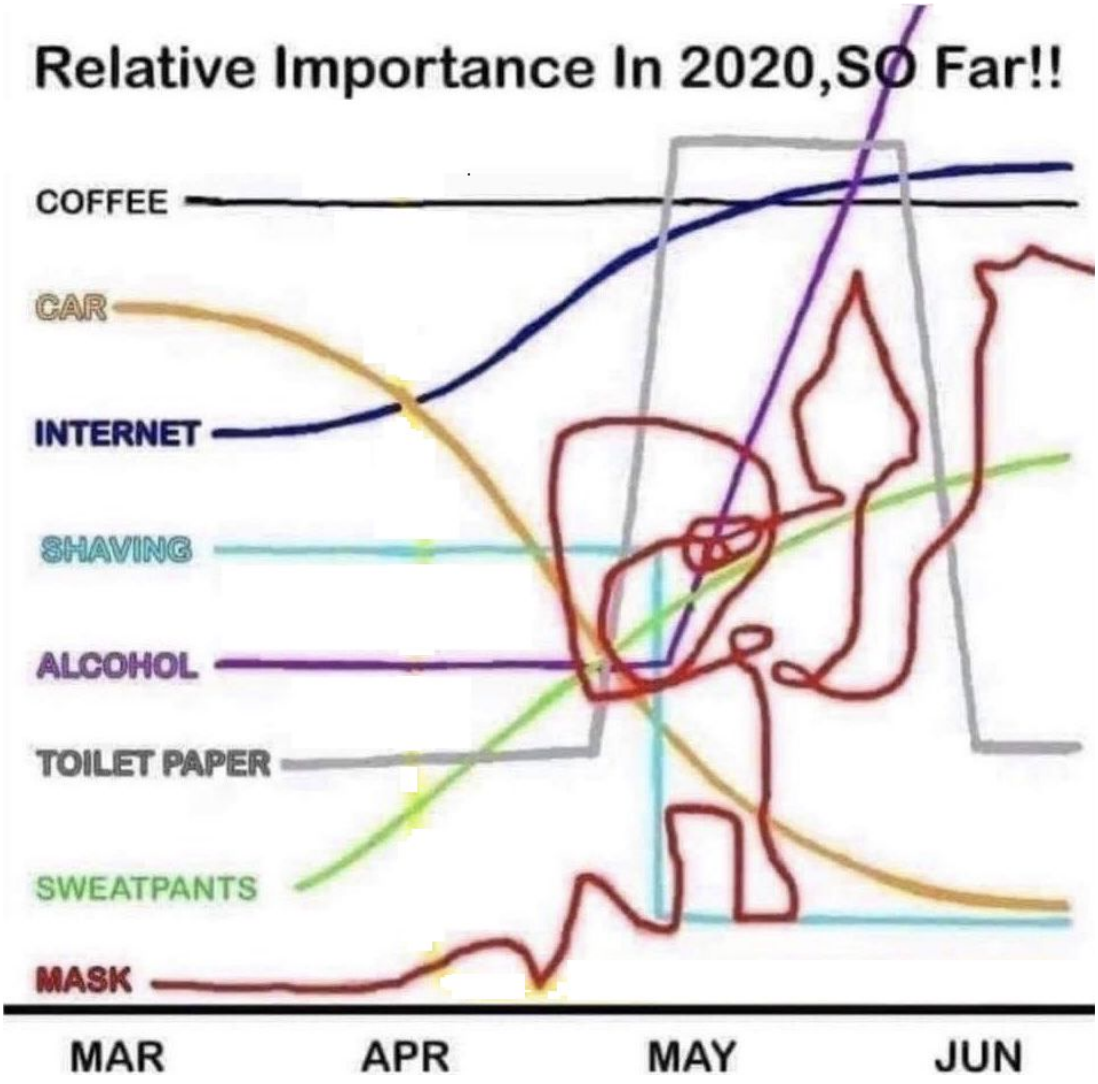
Security Now! #782 - 09-01-20

I Know What You Did Last Summer

This week on Security Now!

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.



Browser News

Chrome 85:

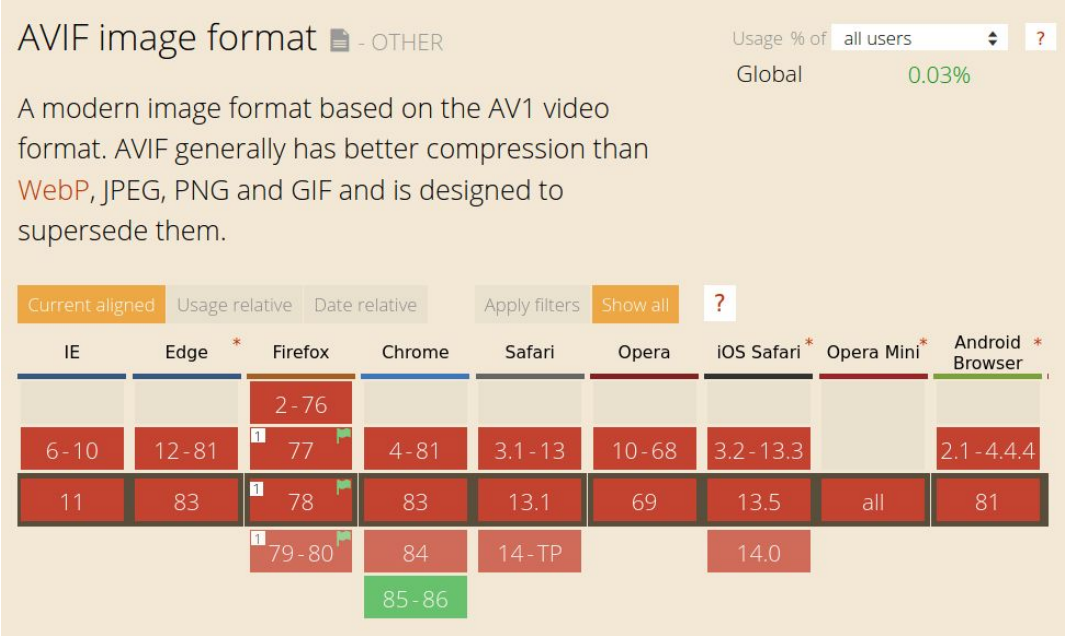
Last week we talked about Chrome 85's release that day which would, among other things, fix a worrisome remote code exploit in its WebGL rendering engine. But I said "among other things". So I wanted to take a couple of minutes to enumerate the other major changes that have landed in the public release of Chrome 85.

In this day of massively heavy web pages, speed is paramount, and we've spoken many times about Google's effective measures to speed up everyone's browser experience, going so far as to pioneer new Internet communications protocols for enhanced connection performance.

Chrome 85 introduces a new compiler optimization technique known as PGO — for "Profile Guided Optimization." Independent page loading benchmarks have measured a significant improvement in the browser's overall performance. 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." Essentially, PGO makes it possible for the browser's most performance-critical code to run faster, which translates into a noticeable speed boost.

And two benchmarks have confirmed this. The "First Contentful Paint" perceived load speed metric and the Speedometer browser benchmark. "First Contentful Paint" refers to the interval between the start of a page load and the browser's first display of any of the page's content. That was measured to improve the browser first paint time by up to 3.5%. And the Speedmeter browser benchmark clocked an 11.4% improvement as well as an overall 7.3% improvement in browser responsiveness. So... Not bad for a free update.

AVIF: There's also an emerging new lossy image compression format which outperforms WebP, JPEG, PNG, GIF and is intended to eventually replace them:



And, as if last Tuesday, exactly one web browser supports it... that's right: Chrome 85. Since most of the other browsers now share the common Chromium core, we can expect more widespread support shortly.

The AV1 Image File Format (AVIF) compresses images using the AV1 codec and has been found to dramatically reduce image sizes without a significant loss in quality. In tests conducted by Netflix using AVIF images, they found that AVIF significantly reduces a file's size while retaining high-level image detail. Not surprisingly, image size matters to Google. But get a load of this! Google wrote: "Reduce bandwidth consumption to load pages faster and reduce overall data consumption: AVIF offers significant file size reduction for images compared with JPEG or WebP. Prior to full optimization, Netflix published results on their test set showing ~50% savings vs JPEG across use cases, going past 60% for RGB." (Where all color components have the same color space.) And AVIF also supports high dynamic range (HDR) and sites such as Netflix, YouTube, and Facebook have shown an interest in its use. So... Google decided to build-in support for the file format. We can expect everyone else to follow suit eventually.

For anyone who's interested in learning more about AVIF, I have a link in the show notes to a terrific page from the libre software folks that provides an very good background:

https://libre-software.net/avif-test/

Mixed Content: 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 is any page asset fetched over HTTP from a base page that was fetched over HTTPS. For a long time only "active" mixed content was being blocked and passive content such as images that were perceived as being less worrisome continued to be allowed. This was also likely a consequence of the fact that many sites were continuing to pull some resources over HTTP.

So today, with release 85, Chrome has started displaying a visual warning whenever mixed content audio, video, images such as .png, .gif, .jpg, .mp4, etc. are downloaded. Chrome is also now blocking other mixed content files that are considered unsafe because they can be abused to deliver malware. These are .pdf, .doc, .docx, .xls, and the like. And assuming that this doesn't result in a surprising end of the world as we know it, the next Chrome — number 86 — will be blocking all mixed content downloads. PERIOD.

PWA Shortcuts: Chrome 85 introduces app shortcuts for progressive web apps (PWA) so that users can quickly access commonly used tasks. When enabled, if a user taps and holds a PWA icon or right-clicks on it, the OS will display various tasks that can automatically be launched using that web app. Oh... And Credge (Microsoft's Chromium Edge) 85 also now has App shortcuts as well.

User-Agent: We've talked about the privacy and tracking problems created by the browser's User-Agent string. Like so many features that were designed as an aid to the web's operation, like cookies, the User-Agent string has been abused for tracking web browser users.

The original cool idea of the User-Agent string was that the browser itself and its various add-ons could add their designations — and their version numbers — to the string so that web

servers might become “version number aware” and thereby work around known problems or lack of features with specific versions by serving up slightly different pages depending upon the profile of who was asking.

To combat this, Google had originally planned to enable add a new feature called 'User-Agent Client Hints' into last Tuesday's Chrome 85. But then the Coronavirus happened and Google decided to drop this back into next year. When it does appear, it will simplify the User-Agent string so that it becomes much less juicy as another tracking signal.

However, this feature has been quietly present since Chrome 84... it's just not enabled yet. But it can be switched on by going here: chrome://flags/#freeze-user-agent flag

Once the "freeze user agent" flag is set, the User-Agent string will be set to historical state for compatibility reasons (presumably it'll say something about Mozilla as all browser do) and it will also show other sanitized details.

Security News

It would have been an inside job

So I'll tease this story by quoting Marcus Hutchins’ Twitter reaction upon learning of it. Just to remind everyone, Marcus is the well known security researcher and reformed cybercrime hacker whose future became uncertain when the FBI grabbed him in Las Vegas' Logan Airport as he was preparing to depart the US for his home in the UK following the annual Black Hat and DefCon conferences. Last Thursday, reacting on Twitter to the news of this story which had just broken, Marcus quite correctly observed: “*One of the benefits of cybercrime is criminals don't have to expose themselves to unnecessary risk by conducting business in person. Flying into US jurisdiction to have malware manually installed on a company's network is absolutely insane.*”

Okay, so what was that all about? Get a load of this: A 27 year old Russian national by the name of Egor Igorevich Kriuchkov, traveled to the US and attempted to subvert and bribe an employee working at Tesla Corporation's massive Nevada Gigafactory. Egor ultimately agreed to pay the employee one million dollars to plant malware inside Tesla's network.

Happily, the employee reported the offer to his employer, Tesla and then worked with the FBI to build an airtight case and set up a sting which included having him covertly record face-to-face meetings discussing the proposal.

In their complaint, which followed Egor's arrest and arraignment, 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 could 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 (Confidential Human Source #1). But last Thursday, Elon Musk confirmed that, yes indeed, it was his company that was the target.

The charging document, which was filed in federal court in Nevada, detailed an extensive and determined attempt to infect Tesla's network.

The defendant, 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 7 in a car Kriuchkov had rented. Referring to the employee as CHS1, prosecutors described it this way:

> 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 (DDoS attack) would be successful for the "group" but the Victim Company's security officers would think the attack had failed.
>
> KRIUCHKOV again listed prior companies the "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. KRIUCHKOV also told CHS1 the "group" had technical staff who would ensure the malware could not be traced back to CHS1. In fact, KRIUCHKOV claimed the group could attribute the attack to another person at Victim Company A, should there be "someone in mind CHS1 wants to teach a lesson."
>
> During the meeting, CHS1 expressed how concerned and stressed CHS1 had been over the request. CHS1 stated if CHS1 were to agree to install the malware, CHS1 would need more money. KRIUCHKOV asked how much, and CHS1 responded US $1,000,000. KRIUCHKOV was sympathetic to the request and said he understood, but would have to contact the "group" before agreeing to the request. KRIUCHKOV confided that the "group" was paying KRIUCHKOV US $500,000 for his participation in getting CHS1 to install the malware, and he was willing to give a significant portion of the payment (US $300,000 to US $450,000) to CHS1 to entice his involvement.

CHS1 said CHS1 would need money upfront to ensure KRIUCHKOV would not have him install the software and then not pay him. Again, KRIUCHKOV asked how much, and CHS1 responded US $50,000. KRIUCHKOV said this was an acceptable amount and a reasonable request but he would have to work on this because he only had US $10,000 with him due to U.S. Customs restrictions on the amount of money he could bring into the country. KRIUCHKOV also questioned what would prevent CHS1 from taking the up-front money and then not following through on installing the malware. CHS1 stated CHS1 was sure KRIUCHKOV or the "group" would figure a way to apply leverage against CHS1 to ensure CHS1 held up his end of the arrangement. CHS1 and KRIUCHKOV discussed the timing of the next meeting, and KRIUCHKOV said he would return to Reno on or around August 17, 2020.

So… Yikes. This would have been a classic inside job. And it serves as a reminder that just as money motivates the bad guys, it's also a time-honored motivator used to turn someone. And it's also a bit freaky that Egor claimed that they had previously -- with success -- done this same thing to some number of other companies. I would imagine that if news of this arrest gets to those who were complicit, they might not be resting easily. You've got to know that the FBI will be working **hard** to obtain the names of the co-conspirators inside those other victim companies. And those victim companies who likely never suspected an insider, must now be wondering which employee may have betrayed them.

More EMV Standard problems

EMV is the monetary transaction method based upon a more than 2,000-page specification. (Upon hearing that, our long time listeners will be shudder.) It is 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.

Next May, 2021, researchers from ETH Zurich, the Swiss Federal Institute of Technology, whose work we have often covered in the past, will deliver a paper at the IEEE Symposium on Security and Privacy, titled: "The EMV Standard: Break, Fix, Verify" https://arxiv.org/pdf/2006.08249.pdf

The paper has been released in advance, presumably because this is really bad and it cannot be readily patched or fixed. 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 that's designed to prevent abuse, and of the payment transaction limit that's designed to limit the damage in the case of abuse.

The paper's Abstract reads:

EMV is the international protocol standard for smart card payment, used in over 9 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 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 application and successfully demonstrated this attack on real-world payment terminals.

Second, 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. *[It would have defrauded the merchant, though it seems like that could have been handled by engaging a willing and interested merchant.]*

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.

The EMV standard, developed by another closed group (much like the WiFi Alliance with all of its many mistakes) also has a long and troubled history of security compromises. To establish a bit of background, the researchers note:

EMV, named after its founders Europay, Mastercard, and Visa, is the worldwide standard for smartcard payment, developed in the mid 1990s. As of December 2019, more than 80% of all card-present transactions globally use EMV, reaching up to 98% in many European countries.

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.

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 (MITM) attacks, card cloning, downgrade attacks, relay attacks, and card skimming are all examples of successful exploits of the standard's shortcomings.

The MITM attack is believed to have been used by criminals in 2010–11 in France and Belgium to carry out fraudulent transactions for 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. *[Meaning, it's too late to fix them now.]*

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, cardholder verification methods, and online/offline authorizations. This raises the question of how we can systematically explore all possible flows and improve the standard to avoid another twenty years of attacks.

So that's what they did. They explain:

In this paper we focus on weakness of 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 is as follows:

1. Bank accepts terminal-accepted transactions: No transaction accepted by the terminal can be declined by the bank.

2. Authentication to the terminal: All transactions accepted by the terminal are authenticated by the card and, if authorized online, the bank.

3. Authentication to the bank: All transactions accepted by the bank are authenticated by the card and the terminal.

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 the bank share a symmetric key that is only known to them.

Using our model, we identify a critical violation of authentication properties by the Visa contactless protocol: 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 application 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 (e.g.,a mobile phone). 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 how does this work in practice?

They use two standard Android smartphones each running a standard custom app. The smartphones need not be hacked. No root privileges or anything fancy required. Just generic custom Android apps which successfully ran on phones from Google and Huawei. The two phones are linked by WiFi. One phone uses its NFC radio to interact with the stolen card, emulating a Point-Of-Sale terminal while the other phone uses its NFC radio to emulate the payment card to the authentic point-of-sale terminal.

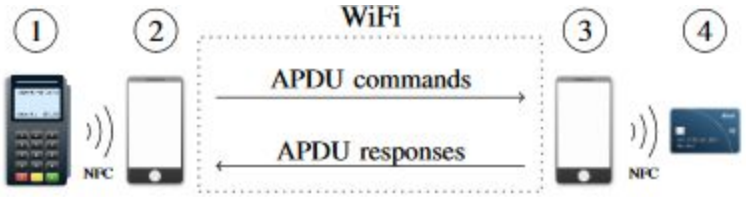


Fig. 3. A relay attack on contactless payment, where (1) is a payment terminal, (4) is a contactless card, and the attacker's equipment are the devices (2) and (3), which are the card emulator and the POS emulator, respectively.

So... in a classic man-in-the-middle attack, the two phones communicate and modify the transaction details on the fly. The point of sale terminal queries the phone being held near it. It relays the query to the POS emulating phone being held near the VISA card. It asks the card to

make a payment, the card does so, then the transaction details are modified on the fly, the modified data is sent back to the smartphone at the authentic POS terminal, which is then empowered to make a very large payment without needing the provide a PIN, because the attacking software has simply modified the unauthenticated transaction data to indicate that a PIN will not be needed and no limits on the transaction amount should be applied.

They conclude their 16-page research paper by writing:

> We have presented a formal model of the latest version of the EMV standard that features all relevant methods for offline data authentication, cardholder verification, and transaction authorization. Using the Tamarin tool, we conducted a full-scale, automatic, formal analysis of this model, uncovering numerous security flaws. These flaws violate fundamental security properties such as authentication, and other guarantees about accepted transactions. We also used our model to identify EMV configurations that lead to secure transactions, and proved their correctness.
>
> Our analysis revealed surprising differences between the security of the contactless payment protocols of Mastercard and Visa, showing that Mastercard is more secure than Visa. We found no major issues with the Mastercard protocol version running in modern cards. Our analysis revealed only minor shortcomings arising from older authentication modes (SDA and DDA) that seem hard to exploit in practice.
>
> In contrast, Visa suffers from several critical issues. The shortcomings we report on, lead to serious, practical attacks, including a PIN bypass for transactions that surpass the cardholder verification limit. Using our proof-of-concept Android application, we successfully tested this attack on real-world transactions in actual stores.
>
> Our attack shows that the PIN is **useless** for Visa contactless transactions. As a result, in our view, the liability shift from banks to consumers or merchants is unjustified for such transactions: Banks, EMVCo, Visa, or some entity other than the consumer or merchant should be liable for such fraudulent transactions. As part of our analysis, we suggested and verified fixes that banks and Visa can deploy on existing terminals to prevent current and future attacks. The good news is that these fixes do not require changes to the EMV standard itself or to consumer cards currently in circulation and they can therefore be feasibly deployed by software updates.

Miscellany (COVID-19)

https://www.youtube.com/watch?v=s_usIkrVQwE

Back in the early days of the COVID-19 pandemic I shared a few YouTube videos that seemed important. The use of my GRC.SC redirect links allows me to get some sense for our listener's interest in various topics. 5,963 of our listeners visited the first, which was the excellent Ars Technica guide. 6,376 visited the second, which was the Coronavirus explained, and 5,942 visited the third, which was the medical school class in COVID-19. Those numbers place COVID-19 at the top of the historical counts of link visits. So today I have the 4th, which is an **astonishing** video explaining the operation of the three primary technologies employed for testing for the presence of the COVID-19 virus or antibodies. The link is: https://grc.sc/covid4

I believe that aside from being seriously educated about the operation of these tests, in **way** too much detail (so just kind of let it wash over you), 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 genetics, and perhaps some better sense for why it appears that so far, testing for COVID-19 has been seriously botched. The short version is: It is a truly delicate error-prone and error-fraught process that will not succeed in an environment of corner-cutting or rushing to a result. But don't take my word for it... if you watch the video, you'll understand, deeply. **https://grc.sc/covid4**

Closing The Loop

Security Now Replay / @SecurityNowRepl
Hi @SGgrc & @leolaporte - I created a way to easily start listening to any old security now episode, as well as some series. Check it out at securitynowreplay.com. Let me know if you have suggestions for additional Series episodes (I just need episode numbers and series name)
https://securitynowreplay.com/podcasts

I received a DM from a listener saying: "Hi Steve, I was wondering if storing password manager database in google drive or drop box is safe ???"

I suggested that the best thing to do would be first use the 7zip archiver with a strong password. It will scrunch the database way down, and once it's encrypted with a strong password it won't matter who obtains it. We talked about 7zip recentl, noting that they did all of the encryption right, using a strong PBKDF based upon SHA-256 to generate the encryption key. At that point, the password manager that's keeping all of those private is probably the weakest link. :)
https://www.7-zip.org/7z.html

Alan Kopp @alankopp1
"A little puzzled by your recommending Syncthing last week. Are you really ok with it doing it's connecting over the Internet without a VPN?"

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 connections is as strong as a VPN's. If a bad guy were to compromise one of the rendezvous servers, they could send their ID to an endpoint and ask to be connected, but that's the extent of the threat. Anyone seeing a bogus request for a sharing connection would simply click "No."

Also, Syncthing clients are able to use public STUN 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. Since all data is fully end-to-end encrypted between client endpoints, there's no exposure of the relayed data to the relay server.
Jon tweeted via DM...

Do you have a recommendation on a Zinc supplement? My doctor suggested I add a zinc supplement to my vitamin regimen. Thanks for all the health insights you've shared on Security Now over the years. (https://www.amazon.com/dp/B00BSHMNA4/)

Yes, there's only one... or at least one very clear way to choose which one from among many.

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. But, as it turns out, in this instance she can be fooled...

I studied this problem about 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. Which is 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 all of this that I easily could because it fascinates me. Perhaps someday. So here's the bottom line: Most mineral supplements are packaged as a simple salt of the mineral. Zinc Citrate or Zinc Picolinate or Zinc Gluconate. The problem with all of these simple salts of Zinc (or of 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, 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 a two-glycine molecule complex. What's special about this complex is that our digestive systems see it as an organic molecule rather than a mineral. And, unlike any of the salts, it is resistant to dissociation and remains intact in our stomach's acidic low pH environment, allowing 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 our bloodstream.

Albion is not a retailer. They exclusively manufacture the bulk supplement raw material. So you won't find any supplements from them. But you will always find their name and trademarks "Albion" and TRAACS on the supplements made by Doctor's Best, Healthy Origins, or others. So, when shopping for mineral supplements, look for the words "Albion" or TRAACS. I went looking for Jon and found a very nice Zinc Bisglycinate supplement on Amazon.

SpinRite

LOLPANDA / @LolMedias

Replying to @SGgrc

Hi Steve I'm sorry to tweet here but...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 0 and 1, it's magic!!! Trillion thanks!

And for a SpinRite 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 many various PCs. So it appears to be ready for its much larger and broader testing debut. Once this podcast is concluded today, I'll begin integrating the earlier IDE hardware driver 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.

So... I don't have anything yet. Don't go looking for a download link. But we're getting there.

I Know What You Did Last Summer

There are two pieces of research, one conducted eight years ago in 2012 and similar, very closely related research which was presented last month during the USENIX 16th Symposium on Usable Privacy and Security.

The earlier paper from 2012 was titled: "Why Johnny Can't Browse in Peace: On the Uniqueness of Web Browsing History Patterns." It explained its purpose and its findings as follows:

> We present the results of the first large-scale study of the uniqueness of Web browsing histories, gathered from a total of 368,284 Internet users who visited a history detection demonstration website.
>
> [We've previously talked about how this can be done: Since our browsers will color previsited URL links differently from new ones, it's possible for a website and server to remotely probe our browser's site-visiting history by placing test URLs into the DOM and then using the Canvas API to determine their rendered color.]
>
> 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 4 visited websites were uniquely identified by their histories in 97% of cases. 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 contents, 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 pages. Finally, we show that indirect history data, such as information about categories of visited websites can also be effective in fingerprinting users, and that similar fingerprinting can be performed by common script providers such as Google or Facebook.

In other words, this introduces another entire category of tracking signal and/or tracking reacquisition in the event of 3rd-party cookie blocking or deliberate cookie deletion. Our browser histories turn out to serve as a surprisingly powerful browser disambiguator.
And, as I mentioned, that research was followed up on and recently updated by a 3-person team

at Mozilla. Their USENIX paper was titled: "Replication: Why We Still Can't Browse in Peace: On the Uniqueness and Reidentifiability of Web Browsing Histories." They explained their work and their findings as follows:

> 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.
>
> 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 ~52,000 Firefox user/volunteers. Our work replicates the original paper's core findings by identifying 48,919 distinct browsing profiles, of which 99% are unique. High uniqueness holds even when histories are truncated to just 100 top sites. We then find that for users who visited 50 or more distinct domains in the two-week data collection period, ~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 third parties pervasive enough to gather web histories sufficient to leverage browsing history as an identifier.

So... in other words, if we know where your web browser has been, as reflected either in the static history it maintains or even dynamically by arranging to watch where it goes in real time and aggregating its visits -- as an ISP can if it's monitoring our DNS lookups -- or as a pervasive advertiser can by tagging us anonymously wherever we go -- or as Google can with their ubiquitous Google Analytics (which begs the question "who's being analyzed") -- there is sufficient diversity in browsing behavior across a large population to uniquely and persistently identify individual users with surprisingly high confidence.

As we've noted, although the rapidly evolving web API does not directly allow querying of a visitor's history, the feature of visited link coloration can be hacked to inform any remote entity who's able to place content into our browser, whether we have previously visited any given site.

I read through their entire paper and their conclusions are rather discouraging. It becomes very clear that probing the history of our browsers provides a valuable signal to those whose mission in life is to determine who we are. I dislike the idea of flushing my browser history, since I often find that having previously-visited links colored differently is useful. But access to our static histories appears to be less important than watching where we steer our browsers over time. And the Internet is now clogged with companies offering exactly those services by each adding their own little snippet of JavaScript code to most of the pages we visit.

https://hal.inria.fr/file/index/docid/747841/filename/johnny2hotpet-finalcam.pdf
https://www.usenix.org/system/files/soups2020-bird.pdf