Security Now! #729 - 08-27-19

Next Gen Ad Privacy

This week on Security Now!

This week we check-in on Texas, and on the Kazakhstan government's attempt to be their own CA. How did that work out for them? We note a troubling increase in attacks on the open source software supply chain, and on Google's announced plans to add data breach notification to Chrome. We look at a surprising APple iOS v10.4 regression whoops!) and on another Microsoft RDP component in need of updating. Then I will update our listeners on the state of SQRL (another of its documents is completed), and on SQRL presentations past and future. I also have some news from my ongoing file sync journey. And then we'll conclude by looking at some very interesting and promising moves as browser-based advertising matures from the ad-hoc mess it has always been into a privacy-respecting Internet citizen.

Could someone please explain this to me?

I mean… really… how is this even possible?

Security News

Texas Ransomware Update
NPR carried a story, and theirs was the only reporting which caught the major of one of the Texas towns stating that the attackers were demanding $2.5 million dollars for the decryption keys of the machines that were successfully attacked and encrypted.

The number of municipalities attacked was revised down by one from 23 to 22. And what we know is that as of last Wednesday, only two cities have come forward to say their computer systems were affected:

Officials in Borger in the Texas Panhandle, said the attack has affected city business and financial operations. Birth and death certificates are not available online, and the city can't accept utility payments from any of its 13,250 residents. City officials said that "Responders have not yet established a time-frame for when full, normal operations will be restored."

Keene, Texas, a city with 6,100 residents outside of Fort Worth, was also hit, officials announced. That city's government is also unable to process utility payments which suggests that they might be sharing outsourced payment systems. And this appears to be confirmed by Keene's Mayor, Gary Heinrich, who told NPR that the hackers broke into the IT system used by the city and managed by an outsourced company... which he said also supports many of the other municipalities targeted. The mayor said: "Well, just about everything we do at City Hall is impacted."

And it was Heinrich who told NPR that the hackers were demanding a collective ransom of $2.5 million. He told NPR: "They got into our software provider, the guys who run our IT systems. A lot of folks in Texas use providers to do that, because we don't have a staff big enough to have IT in house."

So this is looking less like the mass multi-city coordinated attack that was originally claimed and more like a lucky strike on an IT outsourcing firm which provides networked services to many smaller Texas municipalities. Though that's only my own theory based upon the surprisingly scant reporting so far. And that, too, has been surprising. We did note initially that Texas' large cities had not reported any trouble. So that further substantiates the theory.

Remember that Kazakhstan cert?
Reporting from Kazakhstan was murky and the country's users apparently received mixed messages. Some were told that they would need to install the government's private certificate if they wished to retain Internet access. And some users reported indeed being cut off from the Internet until they had installed the certificate. Whereas others said that they had not installed anything and their service was just fine. We heard that only the area in the capitol city will be affected, and other reporting stated that it was entirely optional. We also heard that Kazakh ISPs were forcing their customers into installing this government-issued root certificate on their devices to regain access to their Internet services. So we don't know what the whole story was... And it very much seems likely that Kazakhstan themselves don't either.

But here's one thing we do know: Google, Mozilla and Apple didn't find any of this confusion humorous, and they took immediate action:

https://blog.mozilla.org/security/2019/08/21/protecting-our-users-in-kazakhstan/

Mozilla's blog posting about this last Wednesday contains some new information, so I'm going to share it. They wrote:

> In July, a Firefox user informed Mozilla of a security issue impacting Firefox users in Kazakhstan: They stated that Internet Service Providers (ISPs) in Kazakhstan had begun telling their customers that they must install a government-issued root certificate on their devices. What the ISPs didn't tell their customers was that the certificate was being used to intercept network communications. Other users and researchers confirmed these claims, and listed 3 dozen popular social media and communications sites that were affected.
>
> [Note: This suggests that perhaps the interception was selective, which would make sense, and it would also explain the inconsistent connectivity reports.]
>
> The security and privacy of HTTPS encrypted communications in Firefox and other browsers relies on trusted Certificate Authorities (CAs) to issue website certificates only to someone that controls the domain name or website. For example, you and I can't obtain a trusted certificate for www.facebook.com because Mozilla has strict policies for all CAs trusted by Firefox which only allow an authorized person to get a certificate for that domain. However, when a user in Kazakhstan installs the root certificate provided by their ISP, they are choosing to trust a CA that doesn't have to follow any rules and can issue a certificate for any website to anyone. This enables the interception and decryption of network communications between Firefox and the website, sometimes referred to as a Monster-in-the-Middle (MITM) attack.
>
> We believe this act undermines the security of our users and the web, and it directly contradicts Principle 4 of the Mozilla Manifesto that states, "Individuals' security and privacy on the internet are fundamental and must not be treated as optional."
>
> To protect our users, Firefox, together with Chrome, will block the use of the Kazakhstan root CA certificate. This means that it will not be trusted by Firefox even if the user has installed it. We believe this is the appropriate response because users in Kazakhstan are not being given a meaningful choice over whether to install the certificate and because this attack undermines the integrity of a critical network security mechanism. When attempting to access a website that responds with this certificate, Firefox users will see an error message stating that the certificate should not be trusted.
>
> We encourage users in Kazakhstan affected by this change to research the use of virtual private network (VPN) software, or the Tor Browser, to access the Web. We also strongly encourage anyone who followed the steps to install the Kazakhstan government root certificate to remove it from your devices and to immediately change your passwords, using a strong, unique password for each of your online accounts.

And that bit about changing passwords is a very good point, too, since any login username and password credentials used on those sites would have also been compromised the ability to intercept everything.

(I'll take this moment to note that this is another of SQRL's benefits: SQRL's authentication is secure even if it is not encrypted. The reason is that usernames and passwords are "passive" authentication whereas SQRL is "active" authentication. We finally decided to make SQRL TLS-only, not because it was required, but because there really wasn't any good reason not to. Some people argued that if SQRL offered HTTP access then that would permit non-HTTPS sites to also benefit from it. But since it's not possible to protect the browser's session cookies without HTTPS, it seemed wrong to provide super-secure authentication to a site that really can't use it.)

In any event, Google and Apple have both followed suit with Mozilla, with Google adding the errant cert's fingerprint to Chrome's CRLSet list (so it IS good for something, if not blocking revoked certs) and Google has also added the cert to the Chromium source code base, so that blocking will filter out to all Chromium-based browsers, notably the Chromium-based Edge.

The mixed-Blessing of "Wide Open" Source projects…

The headline on The Hacker News site was: "Hackers Planted Backdoor in Webmin, Popular Utility for Linux/Unix Servers"

With over 3 million downloads per year, Webmin is one of the world's most popular open-source web-based systems for managing Unix-based systems, primarily servers running Linux, FreeBSD, or OpenBSD. Webmin offers a simple UI to manage users and groups, databases, BIND, Apache, Postfix, Sendmail, QMail, backups, firewalls, monitoring and alerts, and more.

So it was with some concern that, following the surprise and irresponsible public disclosure of a critical 0-day vulnerability in Webmin on August 10th during a presentation at DefCon, the project's maintainers revealed last Tuesday that the flaw was not actually the result of a coding mistake made by the programmers. Instead, it was secretly planted by an unknown hacker who successfully managed to inject a backdoor at some point in its build infrastructure... which had persisted into various releases of Webmin (v1.882 through v1.921) remaining hidden for over a year.

This all began when a Turkish researcher presented a 0-day remote code execution vulnerability in Webmin at DefCon on August 10. He had given the Webmin project NO advance notice of either his discovery or his plan to disclose his finding. Joe Cooper, one of the Webmin project's developers said: "We received no advance notification of it, which is unusual and unethical on the part of the researcher who discovered it. But, in such cases there's nothing we can do but fix it ASAP."

Besides revealing the flaw to the public, the Turkish researcher also released a Metasploit module for this vulnerability to automate the exploitation using the Metasploit framework.

The vulnerability, tracked as CVE-2019-15107, was introduced in a security feature that has been designed to let a Webmin administrator enforce a password expiration policy for other users' accounts. According to the Turkish researcher, the security flaw resides in the password

reset page and allows a remote, unauthenticated attacker to execute arbitrary commands with root privileges on affected servers just by adding a simple pipe command ("|") in the old password field through POST requests.

In a blog post published today, Cooper said that the team is still investigating how and when the backdoor was introduced, but confirmed that the official Webmin downloads were replaced by the backdoored packages only on the project's SourceForge repository, and not on the Webmin's GitHub repositories.

Joe Cooper also stressed that the affected password expiration feature is disabled by default for Webmin accounts, which means that most versions will not be vulnerable in their default configuration, and that the flaw would only affect Webmin admins who had manually enabled this feature. Joe Cooper said: "To exploit the malicious code, your Webmin installation must have Webmin -> Webmin Configuration -> Authentication -> Password expiry policy set to Prompt users with expired passwords to enter a new one. This option is not set by default, but if it is set, it allows remote code execution."

However, another security researcher on Twitter later revealed that Webmin version 1.890 =IS= affected in the default configuration, as the hackers appear to have modified the source code to enable password expiration feature by default for all Webmin users.

An unknown attacker made a subtle change to a Webmin script called password_change.cgi. This change gave attackers the ability to send a command through a special URL that an infected Webmin server would then execute with root privileges. In version 1.890, which had more than 421,000 downloads between June, 2018 and last weekend, the backdoor was turned on by default. On versions 1.90, 1.91, 1.91, and 1.92—which collectively had more than 942,000 downloads—the backdoor was active only when admins changed a default setting that allowed expired passwords to be changed. Backdoored versions were distributed on SourceForge, which is the primary distribution source the Webmin website points to.

These changes in the Webmin source code were introduced into the Webmin codebase sometime around April of last year and were later red-flagged by an administrator later last year. But surprisingly, Webmin developers never suspected that it was not their mistake, but that the code was actually modified by someone intentionally. A Shodan search shows that Webmin has nearly a quarter of a million (218,000) Internet-exposed instances available at the time of writing, mostly in the US, France, and Germany — of which more than 13,000 instances =are= running that default-vulnerable Webmin version 1.890. (And note that since that one is a bit old, this suggests that those Webmin instances are not being kept current.)

The threat intelligence firm, Bad Packets, tweeted that there are several actors now actively exploiting the Webmin vulnerability -- beginning the day after its 0-day disclosure at DefCon. One of the scanning teams is the owner of an IoT botnet named Cloudbot.

The Webmin developers promptly removed the malicious backdoor and released clean versions: Webmin v1.930 and Usermin v1.780. Those latest Webmin and Usermin releases also address a handful of cross-site scripting (XSS) vulnerabilities that were responsibly disclosed by a different security researcher who was rewarded with a bounty.

So, needless to say, Webmin administrators are strongly urged to update their packages as soon as possible!

And RubyGems is in trouble again…
A second backdoor came to light on Monday in 11 libraries available in the RubyGems repository. According to an analysis by developer Jan Dintel, the backdoor allowed attackers to use pre-set credentials to remotely execute commands of their choice on infected servers. The malware included a variety of other capabilities, including code that uploaded environment variables which are often a source for static credentials used to access databases, service providers, and other sensitive resources. The exfiltrated material was sent to a server located in Ukraine.

RubyGems officials also found the malicious code included a miner for cryptocurrencies. Download counts showed the backdoored libraries had been downloaded nearly 3,600 times.

Rest-client versions 1.6.10 through 1.6.13, accounting for more than 1,200 of those downloads, were backdoored by someone who compromised an old developer account protected by a previously cracked password. It's unclear how the other RubyGems libraries were infected.

These recent compromises…
...of both the Webmin and RubyGems libraries are only the latest supply chain attacks to impact the software of the open source community. Most people don't think twice about installing software or updates from the official site of a known developer. But as software and websites have continued to become more difficult to exploit, black hats have increasingly exploited this trust to spread malicious wares by poisoning code at its source.

An increase in the focus upon the open source supply chain first came to light last October when two unrelated supply chain attacks against two open source projects were discovered. The first application was the VestaCP control panel interface, and the other was the "Colourama" package that was slipped into the official Python repository.

A month later, malicious code designed to steal funds from bitcoin wallets found its way into event-stream, a code library with 2 million downloads that's used by Fortune 500 companies and small startups. Officials from NPM (the Node JS Package Manager), the open source package manager that hosted the backdoored software, said the malicious code was designed to target people using a bitcoin wallet developed by Copay, one of the companies that incorporated event-stream into its app.

Then, last March, researchers found that another RubyGems library called bootstrap-sass was also backdoored. And early last month we talked about the RubyGems library called strong_password being backdoored. Like the attack discovered this week infecting the 11 RubyGem projects, the bootstrap-sass and strong_password backdoors used a browser cookie function to give attackers the ability to execute code on infected servers. The strong-password backdoor also interacted with smiley.zzz.com.ua, a domain that bears more than a passing resemblance to the mironanoru.zzz.com.ua domain used in the recent attacks.

All of this raises the spectre of what hasn't yet been discovered. Although closed-source software can and has also fallen prey to supply-side attacks, the numbers suggest that the easier-to-

attack low-hanging fruit for supply chain attacks are the open source projects, in part because many don't make multi-factor authentication and code signing mandatory among its large base of contributors. And we should also consider that the influence can be much more subtle. As we know, the NSA has been suspected of subtly influencing some cryptographic standards in ways that might have given the US's intelligence services an edge.

HD Moore, the network security expert, open source coder and hacker who originated and developed the Metasploit Framework said: "The recent discoveries make it clear that these issues are becoming more frequent and that the security ecosystem around package publication and management isn't improving fast enough."

He continued: "The scary part is that each of these instances likely resulted in even more developer accounts being compromised through captured passwords, authorization tokens, API keys, and SSH keys. The attackers likely have enough credentials at hand to do this again, repeatedly, until all credentials are reset and appropriate MFA and signing is put in place."

Moore said the impact of open source supply chain infections is often hard to gauge because backdoored applications can be included as an upstream dependency by another package.

Consequently, he said: "The way that dependency management tools push for the latest packages by default makes a successful attack in the case of a backdoored dependency even more likely," he added.

Chrome to add Data Breach notification

The genesis of this was Mozilla's partnering with Troy Hunt's "Have I Been Pwned" service to create its Firefox Monitor Data Breach Notification service which checked for the presence of the user's credentials -- their passwords or eMails -- were among those leaked in past data breaches.

So Google first created their "Password Checkup" extension to perform a similar function: checking a Chrome user's supplied credentials against a list of some 4 billion leaked and disclosed credentials.

Then, using the analytics provided by its users' use of the Password Checkup extension, Google conducted a study that concluded that 1.5% of all logins have been compromised in data breaches. The study also showed that 26% of users -- just over one out of four -- who were shown a data breach notification changed their password as a result. Since this study demonstrated that providing notifications of compromised login credentials was beneficial to users, Google is now building this support directly into Chrome.

Once this feature is in place, Chrome will alert users when they are logging into sites with credentials that have been exposed by breaches. For the forthcoming feature to work, a user must first be logged into the browser. While logged in, when the user successfully logs into a site with credentials that have been seen in previous data breaches, Chrome will display a warning and suggest that the user might wish to check their passwords... though it's still unclear exactly what that the "Check Your Passwords" button will do.

iOS v10.4 and then quickly v10.4.1
I was surprised that so much was made of a code regression mistake Apple made in their recently released v10.4. I suppose it was noteworthy both because it enabled jailbreaking of iOS devices and because in some circles finding a jailbreak and jailbreaking are the Holy Grails of Apple iOS hacking.

What happened was that a use-after-free vulnerability, which had been discovered and was then fixed in iOS 12.3 reappeared to everyone's surprise in iOS 10.4. But it was short lived. iOS 10.4.1 was quickly released to fix that mistake.

Microsoft's RDP client for Android needed and received an update.
We've recently been discussing the various concerns over vulnerabilities in the Remote Desktop Protocol (RDP) and in Remote Desktop Services (RDS) -- notable BlueKeep.

But recall that there was also a means for remote desktop servers to "reverse compromise" remote desktop clients. The trouble was the recurring problem with vulnerabilities in "interpretation" and, in this case, RDP clients "interpreting" and inherently trusting the data they receive from remote RDP servers.

As we discussed at the time, the Windows RDP clients turned out to have some exploitable weaknesses in their RDP interpreters. And now we have learned that Microsoft's RDP client for Android also had exploits which Microsoft just patched. So, though the threat is very minimal, since you would need to connect to a malicious server, it's worth updating to the latest Microsoft client if you are a user of RDP on Android.

SQRL

I finished the 3rd of four SQRL documents "SQRL Cryptography"
https://www.grc.com/sqrl/sqrl.htm

Last Thursday's OWASP SQRL presentation was a truly terrific event...

I'm reworking the presentation. Random-access slides didn't really work out.

Dublin, Ireland
"Its full Steve! 300 people. The room fits 200 people seated with lots of standing room also."

Mick Ryan, the event organizer, indicated that there are typically many no-shows, which is why they over-booked by 50%... but it might be a good idea to show up early if you want to get a seat. My full presentation is like a security Now podcast... high-speed and non-stop for two hours. Which would be a long time to stand... though the time does fly!

OWASP Göteborg / @owaspgbg
Yihaa what an amazing response we got for the event with Steve Gibson @SGgrc!!
200 tickets and we're fully booked! As usual you can sign up on the waiting list through Eventbrite.
3:49am · 25 Aug 2019 · Twitter for Android

https://www.grc.com/calendar.htm

Steve's File Sync Journey

I'm still on the trek.

Sync(.com) is 1/3rd the cost of Dropbox and offers native E2EE with Windows and Mac desktop support and iOS and Android mobile… but no native Linux support they. They say it's on their roadmap. It also offers easy login with a browser, strong security, file sharing by link.

My only problem with it is that it has a serious memory leak on my main Win7 PC and I have no idea why. It just chews up RAM. It behaves fine on my other Win7 machine. I'm wondering whether the 130 GB or RAM I have on my main workstation might be freaking it out.

And I've solved the annoyance of not having it sync my other folders by moving them under the main "Sync" folder and creating NTFS Junction Points from the original folder locations to the relocated content. Works beautifully.

But I have also become a fan of SyncThing, which is very impressive for Peer-to-Peer file synchronization. And I figured out how to connect it to the cloud for external roaming access.

So… I'm sure that our journey conclusion podcast will have lots of things for people to explore and consider. :)

Next Gen Ad Privacy

Apple -- who, of course, doesn't depend upon revenue from web ads -- has been moving forward on anti-tracking for some time…

Jun 5, 2017 Intelligent Tracking Prevention
https://webkit.org/blog/7675/intelligent-tracking-prevention/

Mar 14, 2018 Intelligent Tracking Prevention 1.1
https://webkit.org/blog/8142/intelligent-tracking-prevention-1-1/

Jun 4, 2018 Intelligent Tracking Prevention 2.0
https://webkit.org/blog/8311/intelligent-tracking-prevention-2-0/

Feb 21, 2019 Intelligent Tracking Prevention 2.1
https://webkit.org/blog/8613/intelligent-tracking-prevention-2-1/

Apr 24, 2019 Intelligent Tracking Prevention 2.2
https://webkit.org/blog/8828/intelligent-tracking-prevention-2-2/

May 22, 2019 Privacy Preserving Ad Click Attribution For the Web
https://webkit.org/blog/8943/privacy-preserving-ad-click-attribution-for-the-web/

What appears to be happening is that we're approaching an inflection point for the industry. Last Thursday, Justin Schuh, the Director of Chrome Engineering, posted two short overviews about Google's new "Privacy Sandbox" initiative which I want to share, then we'll look more closely into the details.

The first was titled "**Building a more private web**"...
https://www.blog.google/products/chrome/building-a-more-private-web/

> Privacy is paramount to us, in everything we do. So today, we are announcing a new initiative to develop a set of open standards to fundamentally enhance privacy on the web. We're calling this a **Privacy Sandbox**.
>
> Technology that publishers and advertisers use to make advertising even more relevant to people is now being used far beyond its original design intent - to a point where some data practices don't match up to user expectations for privacy. Recently, some other browsers have attempted to address this problem, but without an agreed upon set of standards, attempts to improve user privacy are having unintended consequences.
>
> First, large scale blocking of cookies undermine people's privacy by encouraging opaque techniques such as fingerprinting. With fingerprinting, developers have found ways to use tiny bits of information that vary between users, such as what device they have or what fonts they have installed to generate a unique identifier which can then be used to match a user across websites. Unlike cookies, users cannot clear their fingerprint, and therefore cannot control how their information is collected. We think this subverts user choice and is wrong.

Second, blocking cookies without another way to deliver relevant ads significantly reduces publishers' primary means of funding, which jeopardizes the future of the vibrant web. Many publishers have been able to continue to invest in freely accessible content because they can be confident that their advertising will fund their costs. If this funding is cut, we are concerned that we will see much less accessible content for everyone. Recent studies have shown that when advertising is made less relevant by removing cookies, funding for publishers falls by 52% on average[1].

So we are doing something different. We want to find a solution that both really protects user privacy and also helps content remain freely accessible on the web. At I/O, we announced a plan to improve the classification of cookies, give clarity and visibility to cookie settings, as well as plans to more aggressively block fingerprinting. We are making progress on this, and today we are providing more details on our plans to restrict fingerprinting. Collectively we believe all these changes will improve transparency, choice, and control.

But, we can go further. Starting with today's announcements, we will work with the web community to develop new standards that advance privacy, while continuing to support free access to content. Over the last couple of weeks, we've started sharing our preliminary ideas for a Privacy Sandbox - a secure environment for personalization that also protects user privacy. Some ideas include new approaches to ensure that ads continue to be relevant for users, but user data shared with websites and advertisers would be minimized by anonymously aggregating user information, and keeping much more user information on-device only. Our goal is to create a set of standards that is more consistent with users' expectations of privacy.

We are following the web standards process and seeking industry feedback on our initial ideas for the Privacy Sandbox. While Chrome can take action quickly in some areas (for instance, restrictions on fingerprinting) developing web standards is a complex process, and we know from experience that ecosystem changes of this scope take time. They require significant thought, debate, and input from many stakeholders, and generally take multiple years.

To move things forward as quickly as possible, we have documented the specific problems we are trying to solve together, and we are sharing a series of explainers with the web community. We have also summarized these ideas today on the Chromium blog.

We look forward to getting feedback on this approach from the web platform community, including other browsers, publishers, and their advertising partners. Thank you in advance for your help and input on this process - we believe that we must solve these problems together to ensure that the incredible benefits of the open, accessible web continue into the next generation of the internet.

Justin's second posting, in The Chromium Blog was titled: "**Potential uses for the Privacy Sandbox**" https://blog.chromium.org/2019/08/potential-uses-for-privacy-sandbox.html

Today on The Keyword, we outlined our vision for an initiative aimed at evolving the web with architecture that advances privacy, while continuing to support a free and open ecosystem. In order to work toward that vision, we have begun publishing a series of explainers that are intended to be shared and iterated on across the community. Below, we've summarized each of these early proposals, which we are collectively referring to as the Privacy Sandbox.

User information
First, let's identify how user information is currently used in the ad ecosystem so that we can explore the development of the Privacy Sandbox's privacy preserving APIs.

Ad Selection
One of the most challenging questions is what your browser could do to allow a publisher to pick relevant content or show a relevant ad to you, while sharing as little information about your browsing history as possible. We're exploring how to deliver ads to large groups of similar people without letting individually identifying data ever leave your browser — building on the Differential Privacy techniques we've been using in Chrome for nearly 5 years to collect anonymous telemetry information. New technologies like Federated Learning show that it's possible for your browser to avoid revealing that you are a member of a group that likes Beyoncé and sweater vests until it can be sure that group contains thousands of other people.

Conversion Measurement
Publishers and advertisers need to know if advertising actually leads to more business. If it's driving sales, it's clearly relevant to users, and if it's not, they need to improve the content and personalization to make it more relevant. Users then benefit from ads centered around their interests, and advertisers benefit from more effective advertising. Both Google and Apple have already published early stage thinking to evaluate how one might address some of these use cases. These proposals are a first step in exploring how to address the measurement needs of the advertiser without letting the advertiser track a specific user across sites.

Fraud Prevention
Publishers today often need to detect and prevent fraudulent behavior, for instance false transactions or attempts to fake ad activity to steal money from advertisers and publishers. Many companies, including Google, work to detect and prevent fraud, and that's especially true of ad companies and ad fraud. Some of the tools used to legitimately fight fraud today use techniques that can benefit from using more privacy safe mechanisms. One example is the PrivacyPass token, introduced by CloudFlare for Tor users, which is now moving through the standards process.

Protecting the Sandbox Boundary
Our experience has shown us that removing certain capabilities from the web causes developers to find workarounds to keep their current systems working rather than going down the well-lit path. We've seen this recently in response to the actions that other browsers have taken to block cookies - new techniques are emerging that are not transparent to the user, such as fingerprinting. With fingerprinting, developers have found ways to learn tiny bits of information that vary between users, such as what device they have or what fonts they have installed. By combining several of these small data points together they can generate a unique identifier which can then be used to match a user across websites. Unlike cookies, users cannot clear their fingerprint, and this means that even if a user wishes not to be identified, they cannot stop the developer from doing so. We think this subversion of user choice is wrong. As referenced in May at I/O, we are actively taking steps to prevent fingerprinting. We are proposing the implementation of what we call a privacy budget. With a privacy budget, websites can call APIs until those calls have revealed enough information to narrow a user down to a group sufficiently large enough to maintain anonymity. After that, any further attempts to call APIs that would reveal information will cause the browser to intervene and block further calls. We appreciate you taking the time to read through our early proposals for building the Privacy Sandbox. We understand it is ambitious and can't overstate how important it is that this be refined and improved as a result of collaboration across the industry, including other browsers and publishers. We look forward to hearing your thoughts!

https://www.chromium.org/Home/chromium-privacy/privacy-sandbox

The privacy sandbox proposal consists of a series of explainers we are putting forth for comment from the web community.

- Conversion measurement
  https://github.com/csharrison/conversion-measurement-api
  "Click Through Conversion Measurement Event-Level API Explainer"
  This document is an explainer for a potential new web platform feature which allows for measuring and reporting ad click conversions.

- Trust Token API
  https://github.com/dvorak42/trust-token-api
  "Trust Token API Explainer"
  This document is an explainer for a potential future web platform API that allows propagating trust across sites, using the Privacy Pass protocol as an underlying primitive.

- Privacy Budget
  https://github.com/bslassey/privacy-budget
  "Combating Fingerprinting with a Privacy Budget"

  Current state of affairs
  Browsers have been making changes to how cookies are treated. Blunt approaches to cookie blocking have been tried, and in response we have seen some user-tracking efforts move underground, employing harder-to-detect methods that subvert cookie controls. These methods, known as 'fingerprinting', rely on various techniques to examine what makes a given user's browser unique.

  Because fingerprinting is neither transparent nor under the user's control, it results in tracking that doesn't respect user choice.

  End state to aim for
  Fundamentally, we want to limit how much information about individual users is exposed to sites so that in total it is insufficient to identify and track users across the web, except for possibly as part of large, heterogeneous groups.

  There are several ways to quantify the degree to which each user is partially identifiable from the information shared with third parties, including k-anonymity (where k is the number of other users with identical information), entropy (an information-theoretic measure of uncertainty), and differential privacy (ensuring that aggregated data does not reveal the inclusion of an individual's data in the set). Our maximum tolerance for revealing information about each user is termed the privacy budget.

- FLoC
  https://github.com/jkarlin/floc
  "Federated Learning of Cohorts (FLoC)"
  This is an explainer for a new way that browsers could enable interest-based advertising on the web, in which the companies who today observe the browsing behavior of individuals

instead observe the behavior of a cohort (or "flock") of similar people.

Overview
The choice of what ads to show on a web page may typically be based on three broad categories of information: (1) the site or page irrespective of who is reading it (e.g., "put this ad on web pages about motorcycles"); (2) general information about the interests of the person who is going to see the ad (e.g., "show this ad to Classical Music Lovers"); and (3) specific previous actions the person has taken (e.g., "offer a discount on some shoes that you left in a shopping cart"). This document addresses category (2), ads targeting based on someone's general interests.

In today's web, people's interests are typically inferred based on observing what sites or pages they visit, which relies on tracking techniques like third-party cookies or less-transparent mechanisms like device fingerprinting. It would be better for privacy if interest-based advertising could be accomplished without needing to collect a particular individual's browsing history.

We plan to explore ways in which a browser can group together people with similar browsing habits, so that ad tech companies can observe the habits of large groups instead of the activity of individuals. Ad targeting could then be partly based on what group the person falls into.

Browsers would need a way to form clusters that are both useful and private: Useful by collecting people with similar enough interests and producing labels suitable for machine learning, and private by forming large clusters that don't reveal information that's too personal, when the clusters are created, or when they are used.

A FLoC Key, or "flock", is a short name that is shared by a large number (thousands) of people, derived by the browser from its user's browsing history. The browser updates the flock over time as its user traverses the web. The value is made available to websites via a Client Hint.

- Privacy Model for the Web
  https://github.com/michaelkleber/privacy-model
  "A Potential Privacy Model for the Web"
  Sharding Web Identity

  The identity model of the web has been the implicit result of two interacting browser capabilities:

  - Per-domain state, especially cookies, which let one eTLD+1 maintain a consistent notion of a visitor's identity. This identity extends across top-level sites due to 3p cookies, storage within iframes, etc.

  - In-browser passing of information, among the parties co-occurring on a web page (via mechanisms like shared state in DOM or JS, or HTTP redirects, or postMessage).

This combination has led to widely-shared cross-site identities, and so to an ability to perform web-wide tracking of a person's browsing activity. Global static identifiers (like device fingerprinting, or like PII provided by or covertly taken from the person browsing) also offer an independent path to global identity. Limitations on cookies, fingerprinting, and other browser state all aim to reduce this ability to create or access a global identity.

On the one hand, global identity gives rise to the capacity to weave together a record of much of a person's browsing history, a core privacy concern with today's web. Browsers are well-positioned to take action on this issue, by imposing limits on the underlying capabilities exposed to developers. On the other hand, global identity also plays a substantial role in today's web advertising ecosystem. Browsers that impose limitations on these technical capabilities can directly affect publishers' economic viability and encourage work-arounds, if they haven't provided for the legitimate needs of the ecosystem.

This document describes a way the web could potentially work that would not require cross-site tracking, but would still let publishers support themselves with effective advertising.

We need a dialogue within the web platform community — including browsers and the many stakeholders that thrive in and contribute to the ecosystem — so that we can clearly describe a new identity end state that works for everyone. For browsers, this gives us a framework for evaluating proposed changes, standards, and APIs: what boundaries we need to enforce, and where we should innovate to improve the web. For developers, a clear long-term vision provides stability and predictability of the changes along the way.

Any discussion of a new identity model must answer two specific questions:

- Across what range of web activity does the browser let websites treat a person as having a single identity?

- In what ways can information move across identity boundaries without compromising that separation?

This document offers one possible answer to these questions. The goal is a balanced way forward, dramatically improving web privacy while allowing enough information flow to carefully support key needs of publishers and advertisers.