Security Now! #935 - 08-15-23

“Topics” Arrives

This week on Security Now!

Today, we have a birthday to celebrate. And then I wound up encountering so many interesting thoughts shared by our terrific listeners that once I had written everything that I wanted to say regarding the emergence of Google’s long-awaited Topics system to replace tracking, while still giving advertisers what they need, I’d filled up 18 pages of show notes and ran out of space for other news. So next week I’ll catch up with everything else that’s been happening. But the topic of Topics is, I think, important enough to have most of a podcast for itself!

We just need a jumper here…

Security News

Security Now!'s 18th birthday!

Leo, you and I recorded episode #1 of Security Now! on August 19th of 2005. Today is August 15th of 2023... which means that with today's podcast number nine hundred and thirty-five, we will have finished our 18th year, and next week's podcast will be the start of our 19th year.

When I went back to check the date of that first podcast (which was all of 18 minutes!) I got a kick out of its description. Security Now! episode #1 was titled: "As the Worm Turns — the first Internet worms of 2005" and it description made me shake my head: "How a never-disclosed Windows vulnerability was quickly reverse-engineered from the patches to fix it and turned into more than 12 potent and damaging Internet worms in three days. What does this mean for the future of Internet security?" In eighteen years, so much has changed and so much hasn't.

Thanks to feedback from our amazing listeners, one of the things that's been driven home for me during the past couple of years is how much this podcast means to our listeners and, I suppose, how much it would be missed, at least for a while, if it were to ever end. Now, obviously, it's going to end sometime. Unfortunately, Leo, we're not both going to live forever. When William Shatner — who is currently 92 years old, appears to be in remarkable physical and mental health, and recently took that quite emotional ride into orbit and back — was asked about his secret to long life, he replied simply: "Don't die!" So I'll confess that while my middle name is not Tiberius, I'm doing everything I can to follow the Shatner plan.

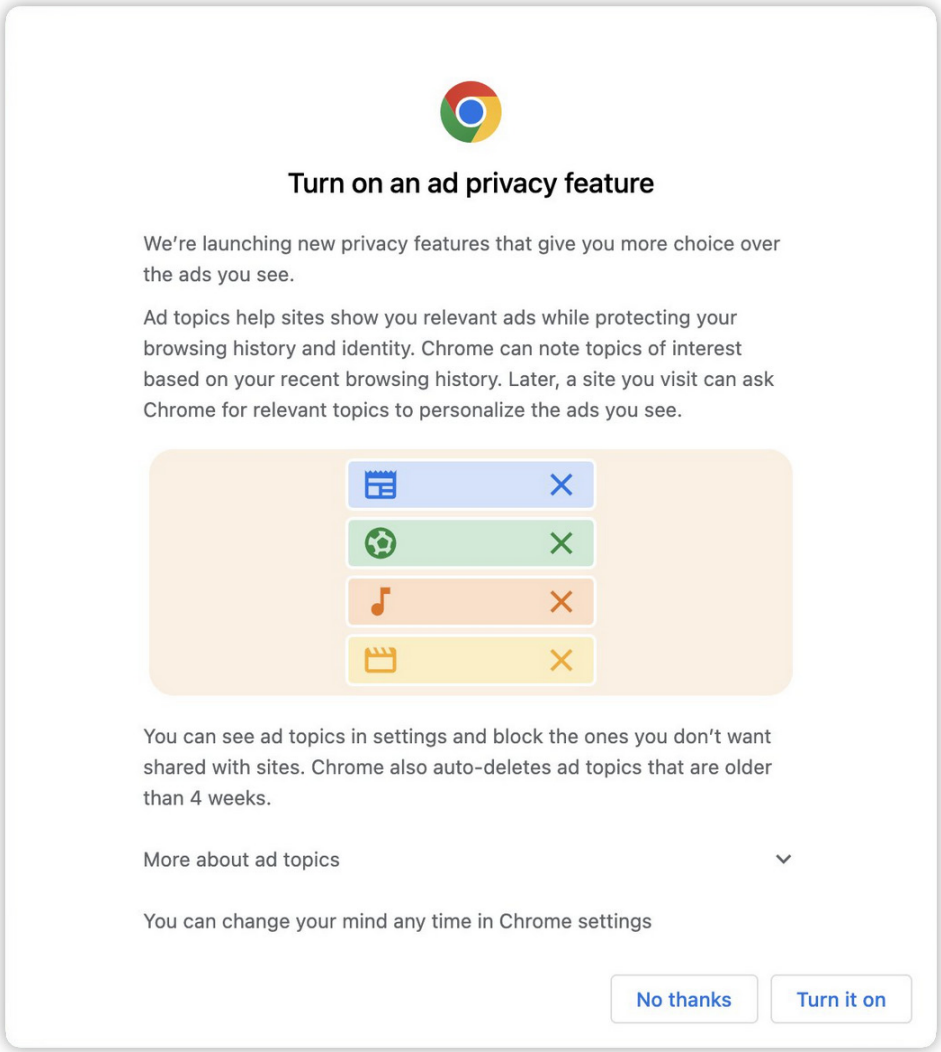
And thanks to our listeners, it occurs to me that perhaps this podcast should follow the Shatner plan, too. As all of our listeners know, I've been talking about ending my involvement with episode #999. I'm here, after 18 years Leo, due to our gentleman's agreement to do a podcast together, and I think that I should remain here as long as that's what everyone wants — which, even after 18 years, seems to be the case today more than ever. If at some point that changes, I'll be happy to recover the time I spend every week producing this. But I feel as though there's a lot of leverage here, by which I mean that this podcast appears to matter to many people, and that's enough for me. So, until something beyond our control occurs, here's to looking toward episode 1,000 and beyond! How far beyond? As we head into year 19, I think we've established that we're not about to run out of material!

And so, on that note, let's begin the final podcast of our 18th year!

Closing the Loop

gimix³ / @gimix3

Hey Steve, today Chrome greeted me with this dialog:



Google adding a privacy feature? And asking me PERMISSION to turn it on? Aha. Extremely suspicious. I guess that's FLoC, which was implemented anyway even if the community pushed back. Any advice on whether we should turn it on or not? Maybe material for the next episode! Greetings from Barcelona, long time listener, Jordi

I received Jordi's note last Friday and I knew what it was. This was not FLoC, this was the long promised roll-out of **Topics**, Google's replacement for hidden, tracking-based profiling in favor of transparent site-based interest profiling which is performed entirely client-side (TNO style) and is entirely under the control of each individual Chrome user. Since I'm no longer a Chrome frequent flier, I hadn't seen the dialog that Jordi shared in this tweet. But I was greeted by it when I opened Chrome yesterday. Since the Chromium browser's **Topics** API is finally making its appearance – and since there's reason to believe that this will be the system that changes the world – it's time to revisit what it is and how it works, which we will do shortly.

Someone who chose the unfortunate handle "burnedeye" sent something very interesting:

burnedeye @burnedeye

Hi Steve! You may find this interesting. I know you're a Firefox user (so am I). Recently, I have found out about this fantastic extension called "Firefox Multi-Account Containers." It solves an issue of being automatically logged in into all Google services when you only want one (for example youtube).

You can isolate youtube.com into its own container where you can be signed in while all the other Google sites such as google.com, gmail etc can stay in a default container where you are not signed in and those sites won't be affected by the fact that you are signed into youtube. It's a browsing tab virtualization in a way. Love it, just as I love the Security Now podcast. Long time fan, Tim.

I was not aware of this slick browser extension and it does solve a problem I sometimes have. It author explains it by writing:

https://addons.mozilla.org/en-US/firefox/addon/multi-account-containers/

The Firefox Multi-Account Containers extension lets you carve out a separate box for each of your online lives – no more opening a different browser just to check your work email!

Under the hood, it separates website storage into tab-specific Containers. Cookies downloaded by one Container are not available to other Containers. You can even integrate individual Containers with Mozilla VPN to protect your browsing and location. With the Firefox Multi-Account Containers extension, you can...

- Sign in to two different accounts on the same site (for example, you could sign in to work email and home email in two different Container tabs.
- Keep different kinds of browsing far away from each other (for example, you might use one Container tab for managing your Checking Account and a different Container tab for searching for new songs by your favorite band)
- Avoid leaving social-network footprints all over the web (for example, you could use a Container tab for signing in to a social network, and use a different tab for visiting online news sites, keeping your social identity separate from tracking scripts on news sites)
- Protect your browsing activity in individual Containers using Mozilla VPN, so you can shop while traveling abroad but check your bank account from a server in your home country.

After installing the Firefox Multi-Account Containers extension, click the Containers icon to edit your Containers. Change their colors, names, and icons. Long-click the new tab button to open a new Container tab.

Though I haven't tried it, this seems very cool. The one thing I would caution is that he writes "keeping your social identity separate from tracking scripts on news sites". But as we know, and as we saw quite vividly last week with the lengths Yahoo!, for example goes to in order to track people, containerizing explicit authentication cookies is not the same as fully anonymizing one's appearance on the web. I'm certain that someone examining the queries being admitted by, or JavaScript running in, adjacent containers would be able to detect that they're running side-by-side in the same browser. Nothing is simpler than noticing that the queries are all emerging onto the Internet from the same private IP address. He addresses this simple IP-based tracking with

the feature of an automatic tie-in with Mozilla's VPN service. But this is not to say that the idea of being able to be logged into the same service under multiple identities with a single browser is not extremely useful. So I wanted to share this listener's tweet so that we all know about it.

Matthew N. Dudek / @mndudek

Hi again, have a question about Full Disk Encryption on SSD's. If an SSD is already in use in an unencrypted state, is it then impossible to fully encrypt it with BitLocker or VeraCrypt due to data stored in inaccessible blocks because of over-provisioning and wear-leveling? How can one encrypt an in-use SSD and guarantee all data is sufficiently scrambled? Thanks again!

Since I plan to fork the early work on SpinRite 7 into a separate product called "Beyond Recall" I will eventually acquire a great deal of firsthand experience solving these problems. Matthew is correct in assuming that wear leveling, defective region sparing and over-provisioning inherently take data-containing mass storage regions out of service rendering them inaccessible through the mass storage device's normal data API. There are two different means for fully erasing all traces of data, even data that's inaccessible. The problem is that they erase all traces of data. What Matthew wants to do is to take an SSD that's already seen some use and add external full disk encryption to that existing device while not leaving the inaccessible regions, which might still contain unencrypted data, unencrypted. And as far as I know, that's not possible. He'd like there to be some command to destructively and permanently erase only all of the inaccessible regions. But I'm not aware that any such command exists. This means that he would need to copy the drive's contents to another device, then arrange to securely erase the entire drive – which would and does include all user-inaccessible areas – then implement BitLocker or VeraCrypt encryption, then restore the drive's original contents.

Trevor Welch / @TWelch333

Hi Steve, I have a spinrite question.
I have a mixed array of drives, hard drives, SATA SSDs, M.2 SSDs, and probably even some weird proprietary external hard drives. I'm finally getting my act together and going to be backing them up to a NAS and backing that up to "the cloud," Is there any advantage to running SpinRite on any of these drives before I do this? Or do I run the risk of maybe pushing one of the drives to it's demise while running SpinRite and being unable to then copy all of the information off? Some of these disks are very old (12-15 years maybe) and are in unknown condition as of right now. So I just want to make sure I give myself the best chance of being able to get as much data as I can off of them.

Love the show and excited for SpinRite 6.1 to come out, maybe an announcement on Tuesday? 👀

Well, no announcement today. As always with a project of this size, with as many moving pieces as this has, working to get to the "there's nothing left to be done" state reveals additional things that need to be done. I'm reminded of that old thought puzzle which suggests that it's impossible to actually get to your destination because you first need to get halfway there. Then halfway again, and halfway again, and halfway again, and so on indefinitely, thus unable to ever

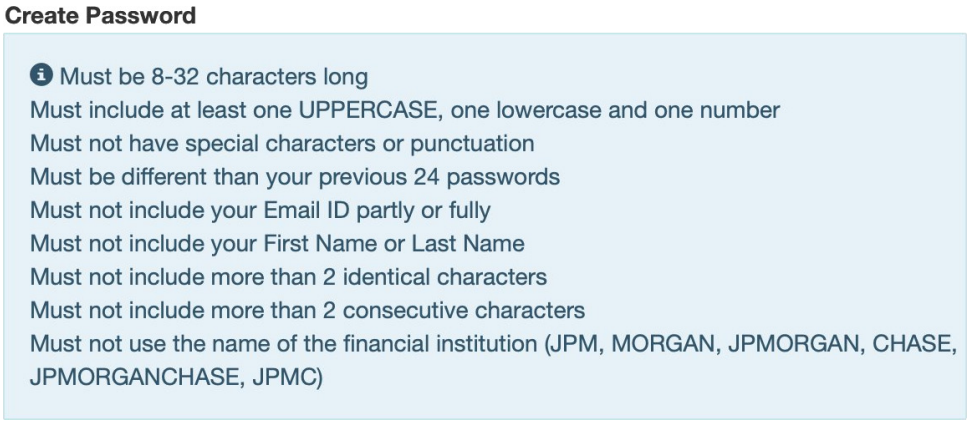
reach the goal. But we are down to very few known things that need addressing. So, yeah, it's looking like one of these weeks soon I'll have an announcement for the listeners of this podcast.

As for Trevor's question, I don't see any reason to run SpinRite on any of those drives ahead of time. Just try copying all of their data off. Copying programs are notoriously finicky about hitting errors during their work, but there are copying utilities that will retry for a while then skip over any trouble they encounter. Even the old XCOPY program from back in the MS-DOS day is still present and it has an "ignore errors" option. And I'm a fan of RoboCopy on Windows. The one thing those tools won't and can't do is deal with problems in the file system's metadata, since they depend upon that metadata to find the file names and file content locations. But otherwise it's usually possible to get most data from a drive.

But if any trouble is encountered along the way I'd say, by all means let SpinRite have the drive to see whether it's able to fix those areas that may have been troublesome.

Matt G / @mpgagne

This might be the most annoying password rule list I have ever seen. You can't use a password generator because any repeat of a single character makes the password invalid. Thought you would enjoy.



Matt's tweet included a screenshot of the password's requirements and they really do require some study... (share them, above).

There are three problems that come to mind: First, one of the rules reads "Must not include more than two consecutive characters." Yet we know from the first rule that the minimum password length is 8 characters. So it's unclear how you create any password longer than 2 characters if you must not include more than two consecutive characters. It must be that the author of this rule meant to say "2 consecutive **identical** characters" except that the preceding rule is "Must not include more than 2 identical characters." ... which means that the misworded following rule is redundant because it's fully covered by the one that precedes it.

But aside from that grammatical nitpicking there are two bigger problems: The very first rule states that passwords must be 8 to 32 characters long. So, a minimum of 8 characters? Really? An 8-character password is sufficient, given all of the rest of the rigmarole that JPMorgan Chase

customers are being put through? And when you think about it, given how difficult they've made it to create **any** password that somehow manages to get through the gauntlet of those rules, users would be hugely incentivised to quit after somehow working out an 8-character string that qualifies.

And that brings us to the third and worst problem of all: These ridiculously onerous rules are going to drive users to create the shortest possible passwords while at the same time making brute force guessing vastly easier and more practical. Any intelligent brute forcer will be informed by the same limiting rules as the password creators. So this dramatically and incredibly reduces the possible brute force search space. *"No special characters or punctuation."* Wow. Talk about dramatically reducing the alphabet size and the search space. Those same rules which make it difficult for a customer to create a qualifying password, automatically discards a vast universe of passwords that would have been possible and that an attacker would have needed to try. But now the attacker already knows that those would not have qualified. Imagine all of the guesses where more than two characters are the same. None of those ever need to be tried.

What a perfect real world example of someone thinking they are being quite clever by forcing their customers into compliance when, instead, they're inconveniencing those customers while at the same time making things far easier for the attackers.

Josh Randall / @JRandall612

*In SN934, you mention that many sites now require you to create an account with an email so they can then spam you later. That is precisely why I use DuckDuckGo's email alias (duckduckgo.com/email) - which I recall you were rather negative about a few episodes back. When I create an account with any new service or site now, I let DDG create a new, random email address for me that is an alias to my main DDG email, which is, in turn, an alias to my actual email. I can receive messages at my actual email through any of these alias'd DDG emails, and reply to those messages, without ever revealing my actual email. And if any site or service ends up spamming me through one of my random aliases, I can simply deactivate it and *poof* no more spam.*

Okay. Let me just be clear about my negativity. It wasn't aimed at DuckDuckGo specifically. I'm 100% behind the use of eMail aliases. I think they rock, and I use them myself all the time. I have hundreds of eMail aliases. But the thing that's different is that the aliases I use are created by my own eMail server at grc.com... which I'm not going to decide to suddenly terminate. Or if I were to, it would be **my** decision under **my** control. Given the importance of eMail as our account recovery and "proof of identity", I would be nervous to be using any 3rd-party provider for such a service. Unlike a use-it-once credit card number, an eMail address needs to be inherently static and persistent. If DuckDuckGo were ever to decide to terminate that service it could create a significant inconvenience for its users who would need to manually change every one of their registered eMail addresses everywhere they had ever used them. Again, I love the idea and I get it that not everyone is able to run their own eMail server, especially since consumer ISPs actively block their customers from running local eMail transports.

David Halliday / @leader4business

In SN933 you highlight that Russians are now prohibited from contributing to open source. However ANY GPL product makes it very clear that the user MUST contribute to the project for the license to be valid. Thus the new Russian Astra Linux based OS can not legally be possible.

As I understand it, the GPL requires that any improvements which are made, thanks to having access to the source code, must be returned to the project. And so, yes, I think that David's point is correct, not that Russia will be particularly concerned about violating the licenses of the West. Russia has essentially stolen Linux from the Linux project; but, with the rising political tensions who's surprised by that?

Thomas Apalenek / @Tom_WA2IVD

Re: Satellite Crowding — Hi Steve. I really enjoyed the episodes and satellite hacking and related satellite info. However, the follow up discussions on swarms of satellites needs to be put in perspective. A 3D graphic of white dots showing all the satellites and debris orbiting the earth does look a little frightening. However, if the satellite size were displayed at the correct scale relative to the earth on the graph, you wouldn't actually see any of the satellites, except for possibly the ISS. Starlink's 40,000+ satellites, in particular, sounds like a lot. But if you imagine 40,000 cars equally spaced over the entire surface of the earth it doesn't seem nearly so bad. The effective surface area of a 300km orbital shell is about 560 million square km. An 1100km orbital shell is over 700 million square km. The Starlink orbital plan includes 3 shells ranging from 340km to about 1100km. The debris issue is a concern, and launch agencies need to make sure it doesn't get out of control. For now, all of the items in orbit are, on average, thousands of miles away from each other. The skies won't be darkening anytime soon.

Thomas, thanks very much for the valuable perspective on this. Very useful.

Brian Norwood / @bnorwood

Catching up on this week's episode and this is what came to mind on the Voyager 2 segment with the shout. Next year we'll find out we let a Borg equivalent know where we are 😂

I think I've noted before, as a Sci-Fi enthusiast, looking at all of the trouble we have right here on mother Earth, where we're all variations on humanity, that as I grow older and hopefully somewhat wiser, I'm coming to appreciate – where I once used to bemoan – how difficult it appears to be to travel between stars. There's that famous and pithy observation that "good fences make good neighbors." So these days I'm much more satisfied with reading fantastical stories in books.

Brian Gluck / @BRGluck

Can you please share the name of the session manager that you use in Firefox? I had one that I loved that saved all the open windows and tabs, but it stopped working a number of years ago. I think it was called sessionrestore, but I honestly don't recall. I would love to know which one you are using, as I miss this functionality and don't like to download an addon into Firefox without a recommendation from someone that I trust.

It's called "Tab Session Manager" as three words and I just noticed that it's also available for Chrome and Edge. And it really is a nice piece of work. I've been using it for many years so I'm happy to vouch for its value and its stability. It's free and user supported. And having just written all that, I just sent its author $25 dollars through PayPal since I'd like to keep it around.

Rusty / @rusty0101

@SGgrc How many people have pointed out that the 2 degrees off that Voyager was pointing at 12.3 billion miles was a bit under 430 million miles, or about 5 times the average distance between the earth and the sun. It's fantastic that they were both able to hear, and send.

Rusty, you take the prize as the only listener whose tweet I've seen who took the time to do the math and give us a calculated answer. My own intuition suggested that this had to be the case, but it's nice to have numbers to go along with that.

"TOPICS" Arrives

The right answer doesn't always present itself the first time. When we encountered "FLoC" – which should stand for "Failed Learning of Cohorts" instead of "Federated Learning of Cohorts" you explained, Leo, that in referring to it as FLoC, Google was clearly struggling to keep with a bird theme. It turns out they were struggling too hard and the FLoC flew the coop. But as I said, the right answer doesn't always present itself the first time. Sometimes it's necessary to experiment and iterate. The design of Topics, which is Google's final solution for the replacement of **profiling by tracking** shows that Google learned a great deal from their previous attempts. And we not only have a system that's ready for prime time, it is beginning to roll out now.

We initially covered the Topics API about a year and a half ago, immediately following Google's first announcement of this proposed new system. This occurred during one of Leo's rare absences from the podcast. Security Now!'s February 1st, 2022 podcast was titled "The Topics API" which was co-hosted by Jason Howell and I always regretted that Leo had missed that discussion since having Leo really understand this is crucial. Fortunately, we have another shot at that, and this time it matters even more since, unlike FLoC, Topics is probably going to fundamentally change the way the Internet works.

As we noted last week, **DNT** stood for "Do Not Track", whereas **GPC** is an explicit request for privacy enforcement. In other words, GPC is not about tracking even though much of the tech press refers to it as an anti-tracking measure. Similarly, Google's Topics solution is not some **new** means for tracking users on the Internet, even though virtually all of the tech press is calling it that. I think the problem is that since tracking is all we've ever known, and change is difficult, everything is assumed to be some form of tracking. But, as everyone is going to understand by the time we're finished here today, Google's Topics system is explicitly and almost painfully a non-tracking solution. If I may be permitted to use the term, it is truly a privacy-forward system for allowing websites to learn a little something about the topics which may interest their visitors. Period. Full stop. Again: "Topics is a privacy-forward system for allowing websites to learn a little something about the topics which may interest their visitors."

It is a means for allowing Google – the Internet's massive advertising behemoth – to continue to deliver user-relevant advertising **without** tracking. Topics does not utilize **any** sort of tracking. None. It's basically Google seeing the writing on the wall that tracking may not be permitted indefinitely – they know that it might eventually be outlawed. So they want to have some sort of user-profiling replacement ready if that happens. And, in fact, they have already announced that Chrome will begin deprecating its support for tracking via cookies starting next year.

I used the term "user-profiling" just now, deliberately, because while that's what Topics is, it is an entirely different and vastly weaker form of profiling from what we're accustomed to. With profiling via tracking, many hidden entities on the Internet know everywhere you go. A visit to a site for erectile dysfunction gets logged into many hidden databases over which we have no control and that information is then made available for sale. It's hardly surprising that's not anything that anyone wants. But under the Topics system, and assuming an absence of tracking which is a separate issue we'll get to in a minute, only your local browser sees that visit. And that visit is not recorded. It only knows that this is a site having the topic of "Health & Wellness",

so that "Health & Wellness" topic gets added to the topics you have shown an interest in for the week. That's it. Thanks to the creation of this new technology that we're going to detail here, Topics provides a means by which a user's web browser may learn by inference about its users' current interests – by virtue of where they take their browser on the Internet. And the browser is then able to judiciously make a few of those interests known to web sites and advertisers who ask.

Now, before we get into the technology, I want to make it clear that, as usual, we're here to talk about technology and that I'm deliberately agnostic on matters relating to the non-technical questions of whether or not **any** profiling is a good thing whether or not it is driven by tracking.

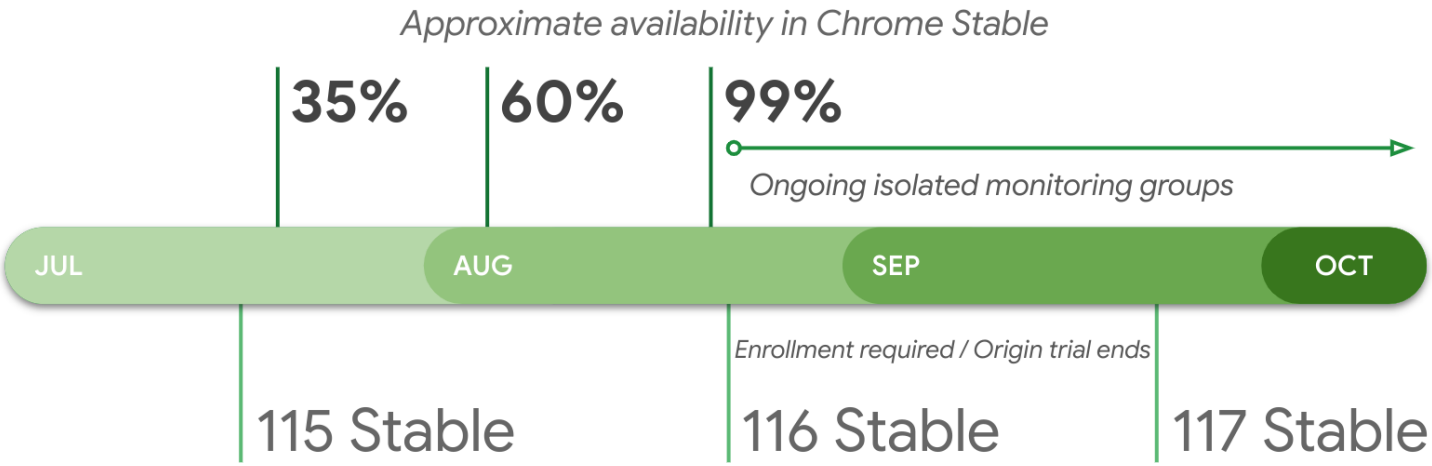
Is **any** form of profiling okay? That question is controversial. There are those who feel very strongly that all use of the Internet should be anonymized to every degree possible. They feel that it is their right to minimize the value they present to a website – and its supporting advertisers – by remaining as anonymous and unknown as possible. They feel that every visit to any website should be siloed, with no other site or third-party content provider having **any** information about them that they don't supply to the site being visited. I get that. Leo and I were just discussing here last week whether the value obtained by user profiling is really worth what advertisers believe it's worth. That's not something I can speak to. Google and advertisers and sites like Yahoo!, clearly believe it is. Or perhaps they just want it because that information is available and they'd like to know who they're spending their advertising money on. Perhaps that feedback allows them to better tune the content that their sites offer. Whatever.

What I do know, though, is that **user profiling via tracking** represents the height of privacy intrusion. As far as I know, an immutable record of every website I have ever visited is squirreled away in multiple massive hidden and inaccessible-to-me profiling databases. And I have zero control over that. That's the world we're in today. But if Topics succeeds, and Google would appear to be in the position to single-handedly deliver its success, it is a **far** less intrusive profiling technology. And in addition to being a much weaker information gatherer, Google has chosen to provide its users complete control over the Topics their browser presents to the world – including turning it off altogether for full anonymity. I'll explain that further in a minute. So, if only on that basis, Topics at least represents a huge step in the right direction. Yes, by default some interest profiling remains. But the means of obtaining those significantly weakened profiles is no longer tracking. And users have complete visibility into their online profile, and are able to curate, edit and even delete it as they choose. Yes, it's a compromise. But there are many websites begging for our support. If voluntarily letting them know something about who we are allows them to generate, as they all claim, significantly more revenue from our visit, is that too high a price to pay? Again, that's an individual decision. But now, at least, it's one we can make.

Another of the arguments presented by the naysayers is that if Topics is embraced we have no guarantee that tracking will end and that we won't merely be adding another powerful profiling technology on top of the existing mix. And those people are right. But as I noted above, Google is planning to begin the deprecation of explicit tracking support via cookies once Topics has come online, and if Google truly wants to prevent tracking once they no longer need neither it nor its bad press, they're in the perfect position to do it. Ultimately, though, as with GPC, it may come down to legislation. And that's fine, if legislation is what's required. Do Not Track might yet be reborn, this time enforced by local or national legislation making it illegal to track anyone who

has indicated that they don't wish to be tracked. And the European Union is likely to outlaw it altogether. If a viable alternative to profiling via tracking is present, as it then will be thanks to Topics, it will be difficult for the trackers to make the case for why users need to be tracked to support the Internet's advertisers and content creators.

So, three and a half weeks ago, on July 20th, TechCrunch noted that **Topics** was finally arriving.



They wrote:

> *Google continues the rollout of its **Privacy Sandbox APIs** — its replacement for tracking cookies for the online advertising industry. Today, right on schedule and in time for the launch of Chrome 115 into the stable release channel, Google announced that it will now start enabling the relevance and measurement APIs in its browser. This will be a gradual rollout, with Google aiming for a 99% availability by mid-August.*
>
> *At this point, Google doesn't expect to make any major changes to the APIs. This includes virtually all of the core Privacy Sandbox features, including Topics, Protected Audience, Attribution Reporting, Private Aggregation, Shared Storage and Fenced Frames.*
>
> *It's worth noting that for the time being, Privacy Sandbox will run in parallel with third-party cookies in the browser. It won't be until early 2024 that Google will deprecate third-party cookies for 1% of Chrome users. After that, the process will speed up and Google will deprecate these cookies for all users by the second half of 2024.*
>
> *The adtech industry has been able to test its readiness for the eventual third-party cookie deprecation, in part through the Relevance and Measurement origin trial. With these features moving into general availability, Google will end this trial (and revoke the tokens to run experiments) on September 20, 2023.*
>
> *For Chrome users, Google will also now start rolling out its user interface to allow them to manage Privacy Sandbox data in the browser, including ad topics, site-suggested ads and ad measurement data. This rollout will run in parallel with the API releases.*
>
> *Google will soon make enrollment and attestation a mandatory process for adtech companies that want to access these APIs on Chrome and Android, though they will be able to continue to do some local testing as well.*

Google notes in their recent announcement: "Shipping these APIs is another key milestone in the ongoing Privacy Sandbox timeline. This marks the beginning of the transition from sites testing in the origin trial to integrating these APIs in production. We will be keeping you updated as we progress through enabling the APIs, to the opt-in testing with labels in Q4 2023, the 1% third-party cookie deprecation in Q1 2024, heading towards the full third-party cookie phaseout in Q3 2024."

Okay. So here's how Topics works: The essence of Topics are individual topic tokens, zero, one or many which are assigned to individual websites. For example, my GRC.com site might be associated with /Computers & Electronics/Network Security and /Computers & Electronics/Programming and /Networking/Internet Security. So when someone visited GRC.com, their own web browser would record their interest in the topics associated with GRC.com. But their visit to GRC.com itself would never be recorded other than in their regular local browser history as is always done. The only thing retained by the browser to indicate their interest in those topics would be those three numbered parameters.

For example, in Google's current 349-topic list, which they refer to as a taxonomy, there's "Arts & Entertainment" as a generic topic if nothing more specific is available. But then there's "Arts & Entertainment" / "Acting & Theater", "Comics", "Concerts & Music Festivals", "Dance", "Entertainment Industry", "Humor", and under Humor is the sub topic "Live Comedy". And it goes on like that with "Arts & Entertainment" having a total of 56 token entries before we switch to "Autos & Vehicles" which has 29 sub categories which brings us to "Beauty & Fitness" and so on. You get the idea. Here's how Google's specification explains this:

The topics are selected from an advertising taxonomy. The initial taxonomy (proposed for experimentation) will include somewhere between a few hundred and a few thousand topics (our initial design includes ~350 topics; as a point of reference the IAB Audience Taxonomy contains ~1,500) and will attempt to exclude sensitive topics (we're planning to engage with external partners to help define this). The eventual goal is for the taxonomy to be sourced from an external party that incorporates feedback and ideas from across the industry.

Under today's tracking technology, if someone were to visit a website concerning the termination of pregnancy, that visit would be tracked and recorded by unknown and unknowable third parties. But under the forthcoming Topics system, the topic associated with that site might be "Health and wellness" and nothing more specific. So this really is privacy forward. And as we'll see, Google has really bent over backwards to ensure that the topics a user's browser volunteers cannot, themselves, be used as a tracking beacon. Google explains:

The topics will be inferred by the browser. The browser will leverage a classifier model to map site hostnames to topics. The classifier weights will be public, perhaps built by an external partner, and will improve over time. It may make sense for sites to provide their own topics (e.g., via meta tags, headers, or JavaScript) but that remains an open question discussed later.

I have a link in the show notes to the official IAB Taxonomy in the form of an Excel spreadsheet: https://iabtechlab.com/wp-content/uploads/2023/03/IABTL-Audience-Taxonomy-1.1-Final-3.xlsx

As a user roams around the web, their browser infers the topics of the various sites they visit using a classifier which maps the site's domain name to its topics. For any given web domain the classifier may return no topics, one or more. There's no set limit, though between one and three would be expected. So it is possible that a site might map to no topics and so doesn't add to the user's accumulating topic history. Or it's possible that a site adds several topics or increases the "popularity weighting" of the user's existing topics.

Web browsers which support the Topics API divide the flow of days into discrete epochs which are each one week long. But it would be more precise to say "one week worth of seconds" since there is no alignment with any calendar. Each browser chooses for itself when each week-long epoch begins and ends. All of a user's browsing activity is grouped into these week-long epochs and only the most recently finished previous three epochs are retained. In other words, as soon as the currently open epoch closes at the end of its weekly cycle, the topics it contains becomes the most recent and is then available along with the two next most recent epochs and the one that had been the oldest of the previous three is discarded.
This has the effect of causing every user's browser to completely forget everything it knew about its user on a rolling basis every four weeks. This is neat, since when a user's interests change the topics which reflect their new interests will realign with them.

In the past, we've talked about the DOM, the standardized Document Object Model. One of the objects that's always present in this model is the "document" object. The document object has properties like the "body" which is the document's body text, "images" which is a collection of all of the images present in the document, and "links" which is a collection of all of the links present in the document. The Topics API adds the "browsingTopics" property to every page's document object.

When a website or one of its advertisers queries the document's "browsingTopics" they will receive up to three topics from the topic taxonomy, **one from each of the preceding three weeks**, -***and-*** those three topics will be returned in random order. When a visitor visits a site, the site will be able to obtain up to three topics which might help it or its advertisers to choose a more relevant ad. It was decided to provide three topics so that a site that doesn't see visitors often will still obtain sufficient information about the user to choose something useful. And a granularity of one week between a user's topic updates was chosen so that sites (and advertisers) which are seen much more often will learn at most one new topic per week.

And not all websites receive the same weekly topic from any given user. Here's how that works:

For each week, the user's top 5 topics which were obtained from the web domains they visited during the preceding week, are determined using that topics classifier — at no time does the user's browser contact **any** external servers for help with choosing; it's all done locally. So, five master topics are chosen from an examination of the preceding week's browsing history. And one additional topic is chosen completely at random from the taxonomy – it will have nothing to do with the user's usage history at all.

Then, when the "document.browsingTopics()" API is called by a site the user is visiting, or JavaScript running in one of its ads, the topic returned for each of the three weeks (which will be returned in random order) is chosen from top 5 plus one random topic as follows: With a 5% chance, so, 1 in 20, the randomly chosen topic will be returned. That's in there as a deliberate wildcard.

Otherwise, the other 19 out of 20 times, the value of an HMAC hash is computed from a static per_user_private_key, the week number, and the document page's website domain name. They're all mashed and hashed together and the result is then taken modulo 5 to produce a uniformly distributed value from 0 to 4, which is used to choose one of the user's five top sites for that week. The same thing is done for the top 5 topics during each of the earlier two weeks and it's those three topics which are returned in random order.

This may seem overly complicated, but it's clever and privacy enforcing: The use of an HMAC keyed by a per-user secret, the week number, and the domain of the webpage, means that for that week, the user's browser will **always** present the same one-of-five Topics (except for that 5% wildcard) to anyone querying **from the same site**, but that every different site will also see an unpredictable but constant one-of-five topics for that user for that week.

This is done to minimize the amount of information being disclosed, since this guarantees that no site will receive more than one fixed topic per user per week. And each site only **ever** receives one of the five real topics which makes it impractical to cross-correlate the same user.

This 5% noise is introduced to ensure that each topic has a minimum number of members, as well as to provide some amount of plausible deniability, meaning that no topic can be regarded as absolute — it may have been deliberately chosen at random. And remember that the exact point in time where a user's week ends and the next one begins is fixed but also chosen at random. So this introduces some additional uncertainty and noise, since not everyone's web browser will calculate new topics at the same time on the same day.

Now, there's one last extremely tricky detail that's a bit of a mind bender. But it's an important privacy safeguard for the entire system. It addresses the problem that FLoC had and which Leo mentioned several times on other TWiT podcasts. The problem was that FLoC was broadcasting a token which was a condensation of the person's web browsing history, and thus by extension a condensation of them. And those who knew how to interpret the token would know what it meant. And this token was being presented to any website they visited without prompting. Leo quite correctly identified this as introducing a significant reduction in user privacy. You'd go to a site you'd never visited before and immediately your browser would be telling them a lot about you.

This problem did not fall upon deaf ears and the Topics API incorporates a mitigation for this. It's worth reminding everyone that there is nothing whatsoever salacious or even really very interesting about the list of topics. They're really quite bland and dry. But they make sense from an advertiser's standpoint. So the first point is there's just no way for anything very personal to be revealed or represented by these topics. But even given that, the Topics API incorporates a very strong topics filter. This is the mind bending part. I'll explain it then provide an example:

Not every website that calls this API **will** receive all of a user's three chosen browser topics. Only API callers that observed the user's browser visiting some site which mapped to the topic in question within the prior three weeks qualifies to receive the topic. In other words, if an advertiser on a website did not call the API sometime in the past three weeks for that user's browser **when** they were visiting a site which mapped to a topic that would be returned now, then the topic will be filtered out and will not included in the 3-topic the array returned by the API. Fewer topics will be returned. Since, as I said, this is somewhat mind bending, here's an example:

During the previous couple of weeks a user has been browsing a lot about travel. So for the time being the browser has learned that about them and chosen to represent their travel-related interest to the world as they visit other sites. So now suppose that they're at a site about gaming and an advertiser on that site runs JavaScript in an ad insertion frame which queries the document.browsingTopics() API to receive three chosen topics of interest to the user—in no particular order—from among the topics that the user's browser has chosen to offer anyone querying about its user while on this gaming site. Remember that each week the top 5 topics of interest are chosen and one of those five is selected from each of the previous three weeks based upon a hash of the domain name being visited.

So... ONLY IF THAT advertiser had queried THAT user's document.browsingTopics API sometime within the previous three week's WHILE that browser was at some other site whose topics matched the topic the user's browser had chosen to offer at this site now, would that topic be allowed to be returned and thus would be presented to that advertiser.

In other words, in order to obtain an interest topic from a user when they are wandering around anywhere on the Internet, an advertiser must have previously asked their browser about them during the past three weeks while they were on a site whose topics may have contributed to the topic they are offering this week. This prevents the problem that FLoC had of recklessly blabbing about a user's interests.

Here's another way of thinking about it: In a world with Topics and nothing else, assuming that we've blocked 3rd-party cookies, fingerprinting and all other tracking, the advertiser doesn't know who the user is. But they will have recently queried the user's browser while the user was at a website whose topics matched the topic the user is now offering. Google explains that this extra topic information filtering is intended to *"prevent the direct dissemination of user information to more parties than the technology that the API is replacing"* — in other words, third-party cookies and other tracking. Another way of phrasing it is that this ensures that 3rd-parties don't learn more about a user's past than they could have with cookies.

As I've mentioned, the history window which limits the inclusion of topics available to each caller is three weeks. Only the topics of sites that use the API, or host ads that query the API, will contribute to the weekly calculation of topics.

Moreover, only sites that were navigated to via a deliberate user gesture are included, as opposed to a redirect, for example. So it's not possible to bounce users around to load-up their browser with topics. If the API cannot be used for any reason, if it's disabled by the user or by a response header, then the page visit will not contribute to the weekly calculation.

If the user opts out of the Topics API by disabling it in browser settings, or is in incognito mode, or the user has cleared their browser history, no topics will be returned.

So, the goal of the Topics API's is to take a step forward toward increased user privacy, while still providing enough relevant information to advertisers that websites can continue to thrive, but without the need for invasive tracking enabled via existing tracking methods.

One other critical aspect of the whole Topics solutions is the user's understanding and sense of control. Very very few users are ever going to understand what we've just described. But they don't need to, because what they see is a list of the topics that their own browser has accumulated over the past three to four weeks. And if they're completely free to either delete any topic they choose or even disable it from ever coming back.

Google wants this to succeed, so they want to give users of their browser control. The API's human-readable taxonomy enables people to learn about and control the topics that may be suggested about them by their browser. And as I said, they're free to delete or disable any of them. And the entire Topics API can be disabled.

And unlike FLoC, which built its hash from every site visited, only sites that include code which calls the Topics API would be included in the browsing history eligible for topic frequency calculations. In other words, sites are not eligible for contributing to topic frequency calculations unless the site or an embedded service has taken the action of calling the Topics API.

Additionally, site will be allowed to block topic calculation for their visitors using a Permissions-Policy header: Permissions-Policy: browsing-topics=(). Any site which returns an empty string for "browsing-topics" the API will be disabled for that site.

No topics will ever be returned if: The user opts out of the Topics API via browser settings, or the user has cleared their topics or even cleared their cookies, or if the browser is in Incognito mode.

So, that's the operation of Google's Topics API which is now-in-place in Chrome and presumably in other Chromium browsers which choose to implement it.

We know what the EFF would say. They want nothing less than pure and absolute anonymity. But website operators say that would cost them dearly. I have no way to gauge how much revenue websites would lose if targeted advertising were eliminated and I certainly wouldn't shed a tear over the end of any companies whose entire business model was secretly tracking users against their wishes.

To me, the Topics API feels as if Google is finally getting really serious about offering an alternative to tracking; one which, apparently, advertisers can live with. Afterall, Google is a massive advertiser themselves and they plan to shutdown Chrome's 3rd-party cookies starting in 2024 with full elimination by the middle of next year. And once they've done that we can imagine that Chrome will become extremely tracking hostile.

Topics gives users not only a sense of control, but actual control, by allowing them to view, delete or disable the topics that are currently being judiciously doled out about them. And this brings us to an interesting question: If our personal interests are separated from tracking, so that the things we're interested in can be shared cleanly, and if that minimal sharing provides significantly greater advertising revenue to the sites we frequent, are we willing to voluntarily give sites that benefit. I know with absolute clarity that I am. If it's really that valuable, please accept it as my micro-payment. The EFF's absolutists may disagree with me. And that's fine. Topics finally makes that an individual personal choice.