The Protected Audience API

Description: What would an IoT device that had been taken over do? And what would happen to the target of attacks it might participate in? What serious problem was recently discovered in a new post-quantum algorithm, and what does this mean? What does a global map of web browser usage reveal? And after entertaining some thoughts and feedback from our listeners and describing the final touch I'm putting on SpinRite, we're going to rock everyone's world (and I'm not kidding) by explaining what Google has been up to for the past three years, why it is going to truly change everything we know about the way advertisements are served to web browser users, and what it all means for the future.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-957.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-957-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. My goodness. What would you do if you found out your washing machine was uploading 3.6GB of data every single day? Why would that be? Well, Steve's got a good solution. We'll find out which browser is now totally dominant in the world. And then we'll find out what Google's doing to protect your privacy and still give advertisers the information they need to target you. Is that possible? Stay tuned. Security Now! is next.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 957, recorded Tuesday, January 16th, 2024: The Protected Audience API.

It's time for Security Now!. Normally Steve would be over my left shoulder here, but he's actually over to my right because - or you meant left. He's over there. He's over there. I'm here in Rhode Island in my mom's house, and visiting Mom, and Steve's at his house. And we are going to do the show from here. But the good news is the quality continues on. Steve Gibson, hello.

Steve Gibson: Yo, Leo, great to be with you, wherever you are.

Leo: Yeah. It snowed here.

Steve: In the snow, where it's already getting dark because you're in the northern latitudes.

Leo: Yeah. You don't want to - yeah. This is - it snowed, and then it rained, and it turned into slush, and now it's going to freeze. And it's just, oh, boy, yeah, arctic chill. You're in beautiful Southern California where it's always perfect.

Steve: I am. And there's no indication that I'll be leaving anytime soon. So I won't have to change any of my cabling. So, okay. Today's topic, today's title, is one of the driest-sounding titles in a while. This is Security Now! podcast 957 for January 16th, 2024, titled "The Protected Audience API."

Leo: Well, that sounds fascinating.

Steve: Begs many questions. What is the audience being protected from?

Leo: Right.

Steve: And what do they need an API for?

Leo: Right.

Steve: Okay. So we're going to explain all that. But first we're going to examine what an IoT device that had been taken over would look like and do. What would happen to the target of the attacks that it might participate in? What serious problem was recently discovered in a new post-quantum algorithm - oops - and what does this mean? What does a global map of web browser usage reveal? And after some entertaining thoughts and feedback from our listeners, and describing the final touch, I think it's going to be final, that I'm putting on SpinRite, we're actually going to rock everyone's world.

Leo: Oh, boy.

Steve: And I'm not kidding, by examining and mostly understanding what Google has been up to for the past three years, why it is going to truly change everything we know about the way advertisements are served to web browser users, and what it all means for the future. And the way we kind of got to this podcast today is odd because I thought I had an idea for what I was going to talk about this week, and I mentioned that I'd had an idea last week. Then when I got into it yesterday I thought, oh, no, this doesn't really - this is not going to work. So the guy was into law. And so but that dragged me into what he was looking at, which was completely, like, what is Google talking about? So then I thought, okay, I can't even talk about that. After I'd invested rather significantly in getting ready to talk about that, I thought, okay, no.

So then I was upset, and I moved it from being what we would talk about into just an item. But then when I tried to sort of massage it away from being our main topic into just a news item, then I thought, oh, I think I kind of understand this. So then I moved it back into our main topic and expanded it further. And it took up pretty much all the air of the podcast. So I'm already tired.

Leo: Just the explanation is exhausting.

Steve: But believe me, this one, as I was writing this, I was thinking, okay. As soon as this thing gets produced and is posted, I need to point Jeff Jarvis at it because this is going to wind him up. I mean, in a good way. He's going to - because, you know, because Jeff likes to understand things, and he keeps telling us how nontechnical he is. Well, everybody's going to understand this, and this is really important.

Leo: Is this the sequel to what Google's doing with killing third-party cookies and, what was it, FLoC and Topics and all the different things they were trying to do to make ads viable without invading privacy?

Steve: Yes. Yes. Yes.

Leo: Okay. So it sounds like you think pretty highly of it.

Steve: It's going to happen. And what is I think the most surprising thing is that the good news is, you know, Tim Berners-Lee is not in a grave from which he could roll over or turn over.

Leo: No. In fact, he's very actively running the World Wide Web Consortium, yeah.

Steve: He's looking great. But what Google has done to their browser - and they did this last summer, this has been active since July of last year - what they have done is astonishingly huge. And by the end of this podcast our listeners are going to understand how the world has changed.

Leo: Interesting.

Steve: And we just haven't woken up to it yet.

Leo: Yeah. Well, this obviously sounds like something everybody should listen to. This is the argument, by the way, and we have this argument a lot on Windows Weekly, about why there shouldn't be just a monoculture with browsing because it gives Google outweighed importance in all of this.

Steve: Well, the good news is that - and of course everything they're doing is open source. So Firefox will end up incorporating this into it. So what Google has essentially - our browsers used to be HTML renderers.

Leo: Right.

Steve: What this turns our browser into is an ad-auctioning server.

Leo: Oh, no.

Steve: It is - I know, Leo. It is huge. It is...

Leo: Oh, boy.

Steve: But it's the only way for Google to deliver what they want and what we demand.

Leo: Okay, good. I mean, this is interesting.

Steve: Yeah. But this is a seminal podcast.

Leo: All right. We're going to have to listen and stay tuned.

Steve: And I'm a little worried about that word "seminal," but, you know, you know how I mean it.

Leo: Just keep the camel's nose out of your kimono, and you'll be good. All right, Steve. I'm ready for the Picture of the Week.

Steve: So, okay. So I've had this one in my bag of tricks for a month or two, just sort of waiting for the right time. And I just love this. So for those who can't, who aren't seeing the picture in live feed, I don't know - we can only see, like, where this object is that is the focus of the picture, not the setting, the larger setting which it's meant to be describing. But we have this large square, it's probably metal, embossed sign where in big huge all-caps, in relief, it says "Please Do Not Touch." So it's referring to something in its environment that we are being told, whoa, do not touch. The punchline here, however, is that that admonition is repeated in Braille below the sign.

Leo: That's something you don't expect to see in Braille is please do not touch, because you're touching it.

Steve: Yeah. So, and I'm wondering what happens if a non-sighted person reaches out and scans this with their fingers. Do they then jump back?

Leo: They leap back. Oh, I'm touching it.

Steve: Because oh my god, you know, I'm not supposed to touch this. Anyway, I gave this picture the caption "Please provide a clear visual example of 'Irony.'"

Leo: Love it. Very nice.

Steve: And Please Do Not Touch in Braille, yeah. Okay. So What would an IoT device look like that had been taken over? That's something we've never talked about. You

know, we talked a lot about the threat that's posed by the remote takeover of IoT devices. We know without any question that there are a great many very large bot fleets, and that they are composed of individual unattended Internet-connected devices.

Well, one of our listeners, Joe Lyon, sent me an image of a Twitter posting where the poster is rhetorically asking why his LG washing machine is using 3.6GB of data per day?

Leo: Wow.

Steve: Yeah, 3.6GB. And he attached an image to his Twitter posting that was produced by some network monitoring tool, showing that something on his network whose interface is labeled "LG Smart Laundry Open" is, indeed, quite busy on the network.

Leo: A little too smart.

Steve: Yeah, exactly, a little too smart for its own good.

Leo: Just wash the damn clothes. You don't need to surf on the 'Net while you're doing it.

Steve: And, you know, whatever is going on is happening very uniformly for a full 24 hours, because this chart that we've got on the show notes shows 24 hours of use with only one hour of the 24 showing a reduced total bandwidth during that hour. So, yeah, there's certainly something sufficient there to raise suspicion. Now, what also caught my eye was that the labels on the traffic flow show a download of 95.75MB like for the day, and a whopping upload of 3.57GB.

Leo: That's not good.

Steve: Now, anyone who's worked with networking gear knows that it's very important to know which directions "up" and "down" are referring to. Cisco has always used - I was very pleased with them about this - the unambiguous terms "in" and "out," as in, traffic flowing into or out of a network interface. So if the interface is facing toward the Internet, then traffic flowing out of it would be up toward the Internet, and traffic flowing into it would be down from the Internet. But if the interface is facing inward toward, for example, connected to a local area network, then the meaning of "in" and "out" would be reversed.

Okay. So without a bit more information about the network's configuration shown in this picture, we can't be 100% certain. But either the washing machine's networking system is badly broken, causing it to continuously download at a rate of 3.5GB of something per day or, as does seem more likely given the evidence, the label "Upload," even though we cannot be certain what that means, suggests that this washing machine has probably become a bot in someone's army. So it's busy doing its part uploading 3.6GB of junk on a continuous basis, presumably nonsense traffic, just causing some remote person grief.

Leo: That makes - now, see, I saw this story, and I thought, well, what could it - is it keeping track of what clothes you're washing? No, it's been compromised.

Steve: Yes, yes.

Leo: Yeah, that makes a lot more sense.

Steve: That would be the conclusion. This is what an IoT device looks like when it's been compromised. So this brings me to two final observations. First, since the typical consumer is not monitoring their local network's traffic in any way, they would have no way of knowing that this was going on at all, ever. And given the closed turnkey nature of an LG washing machine, it's unclear how one would go about reflashing its firmware to remove the bot, even if you knew that one was in there. You know, it might be just living in RAM, in which case pulling the plug, counting to 10, and powering it back up might be all that's needed to flush it out of the system. But then the device might become reinhabited again before long, as we know happens. So the only real solution would be to take the washing machine off the 'Net. Which brings me to my second point. What the heck is a washing machine...

Leo: Doing on the Internet?

Steve: ...doing being connected to the Internet in the first place?

Leo: Exactly.

Steve: You know, is this another of those "just because we can doesn't mean we should" situations? You know, I've owned washing machines my entire adult life. After Mom stopped washing my underwear for me, I took that over. The only thing any of them have ever been connected to is AC power. So is it really necessary for us to initiate a rinse cycle while we're out roaming around somewhere, or to be notified with a message delivered through an app when our clothes are dry?

Leo: But that is the purpose of that. I know I've seen them sell it that way, like you can control your washing machine from anywhere.

Steve: Oh, that's great. So, you know, I get it that if all of that amazing functionality is free and included, and you know these days nothing costs anything anymore, then why not set it up and get it on the Internet? But we're talking about this because of maybe why not to do that. Maybe something has crawled into that machine, and not just because you needed to wash your clothes more often, and set up housekeeping there. Maybe the only thing it's currently doing is flooding hapless remote victims with unwanted Internet traffic. And maybe also, if it wanted to, it could pivot and start poking around inside your residential network.

Leo: Oh, yeah.

Steve: And just maybe that could end up being a high price to pay for the luxury of being notified by an app when the lint filter needs to be changed. So if these sorts of things are going on, like if these sorts of things, these appliances are going to be

connected to your network, again, give some thought to sequestering them on a separate guest LAN which has no access to your high-value LAN. Most of today's consumer routers now offer this feature. That makes it easier to implement than it was back when we first started talking about the idea of LAN separation many years back. You know, remember my "three dumb routers" concept for how to create isolated LANs when that feature was not already built into our routers. Well, the good news is...

Leo: Or better yet, just don't connect it at all; right?

Steve: Exactly. Ask yourself, do I really need - and here's the problem. This was thrown into a washing machine by people who are more concerned about whether it actually gets your clothes clean than it being on the Internet. So the Internet is a throwaway for them. They're not going to be that concerned about the security of their own washing machine that they're shipping. This is not Cisco who is selling you a washing machine; you know? This is LG.

Leo: And probably they have a module they put in all their appliances; right? This is just, you know, the LG Internet of Things module, and we'll figure out how to sell it.

Steve: Right. And it's using code from the dawn of the Internet.

Leo: Yeah.

Steve: Because, you know, it worked, and they don't care. So, you know...

Leo: Unbelievable.

Steve: The end-user needs to care.

Leo: Wow.

Steve: Okay. And speaking of DDoS attacks, this related bit of news was also pointed to by a listener, Sukima, who's at twit.social. He wrote: "I use this service for all of my personal projects, and liked it so much I was motivated to support them financially. And yet they are having a massive DDoS attack and thought it worth talking about publicly, especially as examples of tech doing everything right while still being vulnerable." And in his tweet to me he sent the URL outage.sr.ht. So I went over and took a look. And I wanted to share what I found because it's just such a perfect example. And then we'll talk a little bit more about mitigation strategies.

One of the three guys who runs the service, actually its founder, over at SourceHut, which is the name of the service, he wrote: "My name is Drew. I'm the founder of SourceHut and one of three SourceHut staff members working on the outage, alongside my colleagues Simon and Conrad. As you've noticed, SourceHut is down. I offer my deepest apologies for this situation. We've made a name for ourselves for reliability, and this is the most severe and prolonged outage we've ever faced. We spend a lot of time

planning to make sure this does not happen, and we failed. We have all hands on deck working the problem to restore service as soon as possible.

"In our emergency planning models, we have procedures in place for many kinds of eventualities. What has happened this week is essentially our worst-case scenario: What if the primary datacenter just disappeared tomorrow? We ask this question of ourselves seriously, and make serious plans for what we'd do if this were to pass. And we are executing those plans now, though we had hoped that we would never need to. I humbly ask for your patience and support as we deal with a very difficult situation. And again, I offer my deepest apologies that this situation has come to pass.

"So what happened? At 06:30 UTC on January 10th, two days prior to the time of writing, a distributed denial of service attack (DDoS) began targeting SourceHut. We still do not know many details. We don't know who they are or why they're targeting us. But we do know that they are targeting SourceHut specifically. We deal with ordinary DDoS attacks" - okay. So just that. His phrase "We deal with ordinary DDoS attacks in the normal course of operations." It's like, okay, it's a sad state of affairs that you refer to "ordinary DDoS attacks."

And he says: "And we are generally able to mitigate them on our end. However, this is not an ordinary DDoS attack. The attacker possesses considerable resources and is operating at a scale beyond which we have the means to mitigate ourselves. In response, before we could do much ourselves to understand or mitigate the problem, our upstream network provider null routed SourceHut entirely, rendering both the Internet at large and SourceHut staff unable to reach our own servers.

"The primary datacenter, PHL, was affected by this problem. We rent colocation space from our PHL supplier, where we have our own servers installed. We purchase networking through our provider, who allocates us a block out of their AS" - you know, we've talked about AS numbers, right, autonomous system numbers - "and who upstreams with Cogent, which is the upstream that ultimately blackholed us. Unfortunately, our colocation provider went through two acquisitions in the past year, and we failed to notice that our account had been forgotten as they migrated between ticketing systems through one of these acquisitions. Thus we were unable to page them. We were initially forced to wait until their normal office hours began to contact them, seven hours after the start of the incident.

"When we did finally get them on the phone, our access to support ticketing was restored, they apologized profusely for the mistake, and we were able to work with them on restoring service and addressing the problems we were facing. This led to SourceHut's availability being partially restored on the evening of January 10th, until the DDoS escalated in the early hours of January 11th, after which point our provider was forced to null route us again.

"We have seen some collateral damage, as well. You may have noticed that Hacker News was down on January 10th. We believe that was ultimately due to Cogent's heavy-handed approach to mitigating the DDoS targeting SourceHut." He said "Sorry," and then he said, "Hacker News, glad you got it sorted." Then he said: "Last night a nonprofit free software forge known as Codeberg also became subject to a DDoS, which is still ongoing and may have been caused by the same actors. This caused our status page to go offline. Codeberg has been kind enough to host it for us so that it's reachable during the outage. We're not sure if Codeberg was targeted because they hosted our status page, or if this is part of a broader attack on free software forge platforms."

Okay. So we were just talking about, of course, the LG smart washing machine and the idea that it was apparently sending a continuous stream of traffic, totaling about 3.5GB per day, out onto the Internet for some purpose. So I wanted to put a face on this to

make it a bit more real for everyone. What I've just shared is a perfect example of where such traffic goes, like that this washing machine was apparently emitting onto the Internet, and its very real consequences for people. You know? People are having their lives seriously affected by these sorts of attacks.

Now, Drew used the term "null routing," which is the action taken by major carriers, such as Cogent in this case, when some client, or client's client, or client's client's client, because Cogent is a Tier 1 Provider, is undergoing a sustained attack. They essentially pull the plug. You know, they have no interest in carrying traffic that is indirectly and inadvertently attacking their network. When an attack originates, as most do now, from a globally dispersed and distributed collection of anonymous and autonomous bots, that traffic, which is all aimed at a single common IP address somewhere, will enter the network of a major carrier like Cogent all across the globe, as well. So that means that the attack is crossing into Cogent's routers from all of its many various peering partners who are the ones whose networks have been infected with some bots, or perhaps the traffic is just transiting across their network and originates from some other major carrier's network.

Whatever the case, the real danger of these attacks is its concentration. As the traffic hops from one router to the next, with each hop bringing it closer to its destination, that traffic is being aggregated. It is growing in strength, and it can get to the point of debilitating the routers it's attempting to pass through. This means that the optimal action for any major carrier like Cogent to take is to prevent this traffic aggregation by blocking the attacking traffic immediately at all and each of the many points of ingress and entry into their network from their peering partners.

So Cogent sends routing table updates out to every one of the peering routers on their border, instructing that router to "null route" - meaning immediately discard - any packets attempting to enter their network which are bound for the IP that's under attack. This neuters the attack without causing any harm to their network because it's unable to concentrate. And since there will almost certainly be malicious bots running inside the networks of some of Cogent's client ISPs, this null routing must also be applied internally, as well as on their border.

Okay. But notice that now, with the targeted IP null routed, it's also impossible for any benign traffic to reach its destination service. As Drew wrote, they were unable to even reach their own servers, you know, even if they had some back way into them, because of this null route. No traffic was getting to their servers, good or bad. A major carrier's null routing inherently not only blocks the attacking traffic, but any and all traffic to that service, no matter what. In fact, once the attack has subsided, and full service could be restored, that site will remain dark until someone at the service provider notices that the attack has mitigated and then lifts the network-wide block to allow regular services to resume.

DDoS attacks like this one have become a fact of life on the Internet. Anyone who's working for any major service provider sees them and deals with them now as part of their daily routine. But as we've just seen, this doesn't make such attacks any less significant and potentially devastating to anyone who is committed to keeping whatever services they offer available. And we also know where these attacks originate. They originate from devices exactly like that LG smart washing machine, a gadget that largely operates autonomously where networking is not its primary focus. It was tacked on, as we said earlier, as a feature; so it never got the attention that it needed to be a truly secure networking device.

And we also know that the phrase, unfortunately, "truly secure networking device" almost needs to be said with tongue in cheek because, sadly, it's become an oxymoron. You know, truly secure networking device. Well, it's almost become the holy grail.

Everything we've learned is that it is truly difficult to create and maintain a truly secure networking device. And the more features are added, the more quickly the challenge grows.

Leo: All right. This is really a good episode. You get so much juice in this. But wait a minute. Quantum crypto problems? That's - wait a minute, now.

Steve: Yeah. Okay. So BleepingComputer recently reported the news that many implementations of the already-in-widespread-use post-quantum key encapsulation mechanism known as "Kyber" - which as I said is in use, for example, by the Mullvad VPN and to provide Signal's post-quantum redundancy, we talked about that before - they jumped right on it, and we said, yay, great. But whoops. So it's been found to be subject to timing attacks. The set of attacks have been dubbed "KyberSlash."

Okay. Now, the first thing to understand here is that nothing has been found wanting from the post-quantum Kyber algorithm itself. As far as anyone knows, Kyber still provides very good quantum resistance. The problem and vulnerability is limited to some of the actual code that was used to implement the Kyber algorithm. And this is part of the typical shaking out process that new algorithms undergo. First we need to get the theory right. Then it's tested to prove that it does what we thought. Next, the code is implemented into libraries where it can actually be used and tested in the real world. And it was at this point in the process that these troubles arose.

The problem is that the vulnerable algorithms perform an integer division instruction where the numbers being divided are dependent upon the algorithm's secret key. Whoops. Since division is an instruction whose timing can vary widely based on the binary bit pattern of the specific numbers being divided, this naturally results in a situation where the secret that needs to be kept has the potential to leak out through a timing side-channel. And that's exactly what happened here.

Now, it's such an obvious and well understood problem that it's kind of shocking that this could happen. And really, whoever wrote that code should be scolded a bit. You know, perhaps they were just having a bad day. Or perhaps they're solely focused on theory and not enough on practice. Who knows? But in any event, the problem was found, and it's well understood. And many of the libraries that implemented the vulnerable reference code have now been updated, and more are being updated. So it's a good thing that we're doing this now rather than two years from now, or 10 years from now, or whenever it might be that we actually become dependent upon the strength of these post-quantum algorithms to protect against quantum-based attacks.

We're okay, you know - and to their credit, Signal, remember, added this to their existing crypto rather than switching over to this, recognizing that its unproven nature meant that it really couldn't be trusted fully yet. So Signal was never in danger, and now they're less so.

Okay. And Leo, we've got a cool picture here. And this is apropos of the podcast's main topic. StatCounter produced this somewhat bracing screenshot of global web browser use 12 years ago, back in 2012, and two years ago in 2022. So since today's podcast is all about Google and their Chrome browser, you know, yikes.

Leo: Oh, this is not good.

Steve: So what I want to know is what's going on in Iceland. I think that's Iceland, you know.

Leo: It's gray. What is gray?

Steve: Yeah, well, that's Safari.

Leo: Oh. Wow.

Steve: But, so okay. So for our listeners who can't see because they're listening, thus listener, anyway, the first picture from 2012 shows, you know, what the world looked like, what, 12 years ago. All of the U.S. and Canada and Alaska and Iceland and sort of the northwest of Africa, looks like all of Australia, anyway, all of that is like a blue, and that was IE.

Leo: Yes.

Steve: Everybody 14 years ago was using Internet Explorer. Interestingly, Mozilla's Firefox was scattered around.

Leo: Yeah, look at France.

Steve: In Europe, exactly, Europe and Italy and...

Leo: Asia.

Steve: Yeah.

Leo: Yeah.

Steve: So there was a lot of Firefox use. And of course Chrome was there. It looks like Africa, the whole continent, most of it, except for a little bit of IE, was Chrome. Russia was all Chrome. And there was also some scattered bits of Opera. Anyway, so that was then. Whoa. Take a look...

Leo: That should be the title of the show, Scattered Bits of Opera.

Steve: Little Bits of Opera, yeah. Okay. Now...

Leo: Holy cow. This is depressing.

Steve: On the key of this second updated chart from two years ago, there is blue for Microsoft Edge. I don't know where it is on the map. Literally on the map.

Leo: I don't see any, yeah.

Steve: I don't see any blue. All I see is Chrome.

Leo: Christ.

Steve: Chrome has taken over.

Leo: Is that Iceland or Greenland? What is...

Steve: Oh, good question. Anyway...

Leo: Oh, there's the blue. We found the blue. It's Chad. I don't know where that is.

Steve: This could also be a map of COVID, unfortunately.

Leo: Unfortunately. It's all green.

Steve: You know? So Chrome is the COVID of browsers. It's just - it's everywhere. Okay. So with that in mind, we will be talking about what Google has done to browsers, which is to say all browsers because they're pretty much one and the same.

I'm going to - I have a couple little bits of feedback from our listeners I want to talk about, and then we're going to plunge in. So of course there were predictably many replies from our listeners about my follow-up discussion last week of the Apple backdoor. I just grabbed one to finish out this subject.

David Wise wrote: "Hey, Steve. Listening to your podcast about the Apple vulnerability. Could this be a supply chain hack with the phones being built in China?" My answer? Sure. Absolutely. Unfortunately, literally anything is possible. I think it's safe to say that by the nature of this we'll never have a satisfactory answer to the many questions surrounding exactly how or why this all happened. All we know for sure is that backdoor hardware exists in the hardware that Apple sold and shipped.

And notice by David's question that plausible deniability exists here. All of the several possible sources of this can claim absolute surprise and total ignorance of how this came to pass. Is it significant that it first appeared in 2018, three years after the famous high-visibility 2015 San Bernardino terrorist attack, several years being the approximate lead time for a new silicon design to move all the way through verification, fabrication, and into physical devices? Again, we'll never know. And, yeah, that's annoying.

Osyncsort tweeted: "Hi, Steve. I'm a fan of the podcast and all the great work you've done in your career. Especially a fan of SQRL. Thank you for this great work, and I wish it to be mainstream in the near future." Well, so do I, but don't hold your breath. "I

recently listened to Episode 885." 885, okay, a while ago. "And you briefly touch on a subject that I've been contemplating, getting into infosec. I'm currently thinking about getting into infosec as a career. I'm in my 40s and wanted to know, from your perspective, if 40s is too old to get into the field.

"My career is online marketing, and I've been fortunate to have been doing it from the early days of Web 1.0 to what is now referred to as Web 3.0. However, after COVID, I have not had the same opportunities in the marketing world. So I find myself looking for a new career and thinking infosec may be the solution. Any advice/opinion is welcomed. Thanks in advance."

Okay. So the good news is that there is a huge and unmet and even growing demand for information security professionals today. I think that the trouble with any sort of "am I too old" question is that so much of the answer to that depends upon the individual. A particular person in their early 30s might already be too old, whereas someone else who's 55 might not be. But speaking only very generally, since that's all I can do here, I think I'd say that someone in their 40s probably spans the range where the question of "too old" might start to be an issue if it's a concern for them. Early 40s, not so much. Late 40s, well, maybe a bit more so.

But regardless, there's definitely something - a lot, actually - to be said for having some gravitas and life experience that only comes with age. An IT guy who's more world-wise will generally be much more useful than a fresh newbie who is still addressing the world with impractical expectations. And especially for an IT guy, knowing how to talk to others is a skill that should not be undervalued. So I think that on balance I'd say go for it, and know that the demand for that skill set will only be increasing over time.

Leo: This is where I would put in the plug for ACI Learning, our sponsor, or ITProTV. And, you know, you could certainly learn, I don't think it's ever too late to learn the skills.

Steve: Right.

Leo: So really the only question you're asking is can I get hired. And, you know, that's in the pan if you've got the skill set, if you've got the certs. I think it's...

Steve: And Leo, if this guy has been listening to this podcast for, like...

Leo: Yeah, that's a good way to start.

Steve: From the beginning.

Leo: Yeah.

Steve: We keep hearing from people, it's like, yeah, I just went in, I didn't even study, and I passed the test.

Leo: Good training. Yeah, yeah.

Steve: It's like, okay, that's great.

Leo: Yeah, that's a good point.

Steve: So someone whose handle is 3n0m41y, which gave me some pause until I realized it was supposed to be anomaly, he said: "Hi, Steve. I would like to get your opinion on Proton Drive versus Sync as a secure cloud storage network. Recently the iOS Sync app has broken the ability to natively use the built-in iOS Files app to navigate Sync's folder structure properly. What happens is that after drilling down one to two directories, the Sync app pushes the structure back to the root folder. While this is not a show stopper, it does break the use of other third-party apps on iOS. I've reached out to the Sync dev team, but they've responded that it will take 'quite a while' to fix. This functionality broke about two months ago. So I just want to get your take if Proton has matured enough to be a replacement for Sync. Cheers, and Happy New Year."

Okay. So, first of all, let me just note that I'm very disappointed when something I've deeply researched and then strongly endorse - anyone remember LastPass? - later evolves, or devolves, in such a way that I come to regret my previous full-throated endorsement. So I'm disappointed to learn that Sync is not standing behind and repairing their various client apps in a timely way.

As for Proton Drive, I have not looked deeply enough to form any opinion one way or the other. However, its "Proton" name should be familiar, since these are the same Swiss people who offer the very popular Proton Mail service. So my strong inclination would be to trust them. What I have no idea about is how their feature-rich offering might match up against Sync. But my shooting-from-the-hip thought would be that, if it does what you want, for a price that makes sense, I'd say that based upon their past performance on everything else we know they've done, I'd be inclined to give them the benefit of any doubt. That's obviously not definitive, but at least it's something.

Okay. So, and last a note about SpinRite. It has been so extremely useful these past final months of work on SpinRite to have this podcast's listeners taking the current SpinRite release candidate out for a spin and providing feedback about their experiences. That feedback has been the primary driving force behind the last few improvements to SpinRite 6.1, which turned out to be quite significant. So I'm glad that I did not declare it finished before that. And it's been a slowly growing chorus of feedback about something else that caused me to decide that I needed to change one last thing. Sort of echoing Steve Jobs.

If you've been following along, you'll recall that one of the astonishing things we discovered during SpinRite's development was that all of the original and past versions of the PC industry's most popular BIOS, produced by American Megatrends and commonly known as the AMI BIOS, contained a very serious bug in its USB handling code. Any access to any USB-connected drive past the drive's 137GB point - which is where 28 bits of binary sector addressing overflow into the 29th bit - causes these AMI BIOSes, which are in the majority, to overwrite the system's main memory right where applications typically load.

When this came to light, I was so appalled by that discovery, and by the idea that this could be very damaging, not only to SpinRite's code in RAM but potentially to its users data, that I decided to flatly prohibit SpinRite 6.1 from accessing any USB-connected drive past its 137GB point. The next SpinRite won't suffer from any of this trouble since it will have its own direct high-performance native USB drivers. So my plan was just to get going on SpinRite 7 as quickly as possible. But SpinRite's early users who have attached

larger-than-137GB drives via USB, and then had 6.1 tell them that they could only safely test the first 137GB of their drive, have not been happy.

And also since then, one of the guys who hangs out in GRC's newsgroups, Paul Farrer, whom I've referred to before when this bug was happening, he was curious to learn more about this. So he looked into the problem while I continued to work on other aspects of SpinRite, working towards just getting it done. Paul wrote a set of exploratory DOS utilities and tested them with a bunch of old motherboards owned by some of our SpinRite testers in the newsgroups.

What he discovered suggested that more could be done than just turning my back on all USB BIOS access in disgust. And the disappointment I was seeing from new people being exposed to SpinRite's refusal to work with any USB BIOS convinced me that I needed to fix this. So I started on that last week, and I expect to have it finished this week because it's not a big deal.

Since only the AMI BIOS is known to have this problem, SpinRite will start by lifting this blanket ban from all non-AMI BIOSes. Then for AMI BIOSes, since they don't all have this trouble, I've reverse-engineered the code surrounding the bug, and I now fully understand what's going on. So SpinRite can now detect when the bug is actually present and can patch the buggy BIOS code in place, to raise SpinRite's access limit from 130GB to 2.2TB. The buggy AMI USB BIOS code will still have a bug that prevents SpinRite from working past 2.2TB, but that's way better in today's world than clamping all USB BIOS access for everyone at 137GB. So that's the SpinRite that everyone will get, and maybe next week. So again, a nice feature benefit. And again, I'm glad that I waited and am putting this in.

Leo: Next week, really, next week?

Steve: Like, yeah.

Leo: Maybe.

Steve: Close. I mean, like...

Leo: No promises, please.

Steve: There's really nothing left. I'm really happy with it. Nobody has found any problems at all for the last couple months now and, you know, while I've been like letting it sit and stew and cook. So, yeah.

Leo: That's awesome.

Steve: We're right there.

Leo: Well, we'll have a cake ready for you. We'll have a party. Fireworks.

Steve: Just don't throw it in my face.

Leo: Yeah, well, no, I won't - we might have some confetti that we can throw in the air instead. How about that? A little party for you, Steve.

Steve: That would be very nice.

Leo: All right. We're going to get to the heart of the subject. I can't wait.

Steve: Oh, boy. Oh, boy.

Leo: What is Google doing to protect your privacy and to help advertisers? We'll find out. This is kind of breaking news, I think. So I'm looking forward to hearing this. All right, Steve.

Steve: Okay.

Leo: I'm ready. Let's find out what's going on. What's Google up to now?

Steve: This is big. I mentioned last week that I thought I might be onto an interesting topic to explore this week. It turned out that while the guy I stumbled upon was the real deal, his several blog postings were sharing pieces from his Master of Law dissertation for the University of Edinburgh. After I looked into it more deeply, it didn't really make for the sort of content I know our listeners are looking for. This scholar was carefully examining the legal and policy implications of Google's recent work on the web, the set of new technologies collectively known as "The Privacy Sandbox." And he was looking at it against EU and UK laws like the GDPR, what would it mean in that context.

And this guy was not some lawyer. He is a deep technology guy who has been actively involved with the W3C, serving on many committees and having co-authored a number of web specs. His focus has always been on privacy; and he's the guy who, years ago, realized that the addition of a high-resolution battery level meter into the HTML5 specifications would provide another signal that could be used for fingerprinting and tracking people across the web.

But as I said, his focus was on what Google's recent work would mean legally. And for what it's worth, this very well-informed legal and technical academic, this guy who is also a privacy nut, is quite bullish on the future Google has been paving for us. So that just means that what we are going to talk about this week is all the more relevant and significant.

And what we are going to talk about this week is something known as the Protected Audience API. It's another of the several components which make up what Google collectively refers to as their Privacy Sandbox. Now, the name Protected Audience API is every bit as awkward as many of Google's other names. You know, they're a big company. They could afford to employ someone with the title of Director of Naming Things, and give this person a big office and a staff, because it's clear, and it will soon become much clearer, that the nerds who invent this technology should not be the ones to name it. In this instance, what's "protected" is user privacy, and "audience" refers to

the audience for web-based display advertising. But as it is, calling this the Protected Audience API only tells you what it is after you already know, which is not the definition of a great name.

In any event, this collection of work that Google has called their Privacy Sandbox currently contains a handful, dare I say a plethora, of different APIs. There's the new Topics API which we've previously covered at length. And there's the Protected Audience API which is what we'll be looking at today. But then there's also something known as the Private State Tokens API, the Attribution Reporting API, the Related Website Sets API, the Shared Storage API, the CHIPS API, the Fenced Frames API, and the Federated Credential Management API. And if you didn't already know what those things are, knowing their names only helps with, you know, very broad strokes.

But here is what everyone listening really does need to know: All of this brand new, serious, deliberately user privacy-focused technology which Google's engineers have recently created and somewhat unfortunately named is real. It collectively represents a truly major step forward in web technology. We all grew up in, and cut our teeth on, extremely simple web technology that its founders would still clearly recognize today. You know, even after many years, this baby hadn't grown much, and it was still far from mature. We had cookies and JavaScript and ambition, and a lot of ideas about what we wanted to do with the web. But everyone was heading in their own direction, doing whatever they needed for themselves just to get the job done, and no one was thinking or worrying about longer term consequences.

The web lacked the architectural and technological depth to get us where we wanted to go in the way we needed to get there. So we wound up with the absolute chaos of tracking and identity brokering and personal data warehousing, deanonymizing, and all the rest of the mess that defines today's world wide web. And an example of the mess we're in today is the utter pointless bureaucracy, you know, the bureaucratic insanity of the GDPR forcing all websites to get cookie usage permission from each of their visitors.

We know that Google is fueled by the revenue generated from advertising. Advertisers want to know everything they possibly can about their audience. They want to optimize their ad buys. And users are creeped-out by the knowledge that they're being tracked around the Internet and profiled. And being the super heavyweight that it is, Google is increasingly coming under scrutiny, you know, under the microscope. But they also have the technological savvy, probably unlike most other players on Earth at this time in our history, to actually solve this very thorny problem which arises from the collision of apparently diametrically opposed interests on today's web. One thing is clear: We're in desperate need of more technology than cookies.

Google began the work to truly solve these problems in earnest three years ago, at the start of 2021. And this wasn't some half-baked attempt to gloss over the true problems that are inherent in the use of a system that was never designed or intended to be used as it is being used today. Google's Privacy Sandbox initiative was, and today is, a significant step forward in web browser technology and standards which is designed to allow the web to finance its own ongoing existence and services through advertising, without in any significant way compromising the privacy of its users.

Okay, now, I get it. We've all been so badly abused by the way things have been that it may be difficult to accept that there truly is a way for this to be accomplished. But there is, and Google has done it. In the future, the use of the web will be much more private than it ever has been since it was first conceived. What's required to make this possible is way more technology than has ever been deployed before. What's been done before now couldn't even be called a half measure.

All of the various APIs I mentioned above, you know, whatever it is they each do, became available in the middle of last year at the start of the third quarter of 2023. They are all operable today, right now, have been for the last six months, and they are in the world's dominant web browser and other browsers that share its Chromium engine. And it's not as if there wasn't something, well, some wrong turns that were made along the way; right? But that's also the nature of pioneering where the path hasn't already been mapped out. FLoC, remember, Google's Federated Learning of Cohorts, was an attempt at generating an opaque token that revealed nothing about the user's browser other than a collection of their interests. But FLoC didn't get off the ground.

Leo: It failed. It fell on the ground.

Steve: It was later replaced - yes. It was later replaced by Topics, which is a complex, but extremely clever system for doing essentially the same thing, but in a far less opaque and thus far more understandable fashion. Topics allows the user's browser to learn about the user by observing where they go on the web, all of which information is retained by and never leaves the browser. Then, through the use of the Protected Audience API, which I'll get to, the user's browser is able to later intelligently select the ads that its own user will see. I know. If that comes as something of a surprise, it should, since it's certainly not the way any of this has ever worked before.

Okay. We've got a lot to cover. It's good stuff. One of the key features to note and to keep in mind is that this expands the role of the web browser significantly. There is now far more going on under the covers than ever before. It was once fun and easy to explain how a web browser cookie worked. It wasn't difficult to explain because there wasn't much to it. But there is very little that's easy to explain about how these various next-generation Privacy Sandbox browser APIs function. And this is made even more difficult by the fact that they're all so deeply interconnected.

When we originally discussed Topics, we had no sense that its purpose was to allow the user's browser to autonomously perform ad selection. But that was always Google's intention. We just needed to see more of the whole picture. And even when we were only seeing a small portion of the whole, explaining the operation of Topics required a very careful description because it is laced with important subtleties. And I suppose that's the main point I want to convey here because we're now asking so much from the operation of the web, even wanting things that appear to be in direct opposition, the simple solutions of yesterday will not get us there.

So what is this Protected Audience API? Believe it or not, opaque as even that name is, the good news is they renamed it to Protected Audience API from what it was before. Which of course begs the question, "Renamed it from what?" Okay. Recall that earlier that they abandoned FLoC (F-L-O-C), which stood for Federated Learning of Cohorts. In a similar vein, the Protected Audience API was originally named FLEDGE, and that was a painful, you know, they won't give up on these birds. It was a painful reverse-engineered acronym which stood for First Locally-Executed Decision over Groups Experiment.

Leo: Oh, that's awful.

Steve: Oh, my god, yeah.

Leo: That's really bad.

Steve: Okay, now, not an exactly catchy name. You know, where is the Director of Naming Things when you need them? Because nerds should not name things, clearly. Okay. And what you're really not going to believe is that FLEDGE grew out of a project named "TURTLEDOVE." I kid you not. And yes, TURTLEDOVE was also an acronym, short for Two Uncorrelated Requests, Then Locally-Executed Decision On Victory.

Leo: God. That's terrible.

Steve: It's really bad.

Leo: It's worse. They're getting worse.

Steve: They're only missing a word to provide the "E" at the end of DOVE. So, Excellent? Everlasting? Or maybe Excruciating?

Leo: Yeah.

Steve: Yeah. Anyway, I was able to explain how Topics worked since, while it was a bit tricky and subtle, it was a relatively self-contained problem and solution. I don't have that feeling about this Protected Audience API because, as I noted earlier, they each only really make coherent sense when they're taken as a whole. So I'm not going to explain it at the same level of transactional detail. Okay. But I want to at least share some sound bites so that you can come away with some sense for what's going on here. And believe me, that will be enough.

So at the start of Google's Protected Audience API explainer page, it opens with one sentence that needs to be taken absolutely literally. Okay. They start with: "On-device ad auctions to serve remarketing and custom audiences, without cross-site third-party tracking." Okay. "On-device ad auctions." Wow. Okay, now, I don't expect anyone to understand in any detail what follows. I don't. So just let it wash over you, and you'll get some very useful feeling for what's going on.

Google "explains," and I have "explains" in air quotes: "The Protected Audience API uses interest groups to enable sites to display ads that are relevant to their users. For example, when a user visits a site that wants to advertise its products, an interest group owner can ask the user's browser to add membership for the interest group. If the request is successful, the browser records the name of the interest group, for example, 'custom bikes'; the owner of the interest group, which is a URL, like 'bikes-r-us.example'; and interest group configuration information to allow the browser to access bidding code, and ad code, and real-time data, if the group's owner is invited to bid in an ad auction."

Okay. I know. Now, just let your head spin. It'll be okay. So there is a feeling of the way Topics works here. The key is that the user's browser visits a site like "custom bikes." And because their browser is at that site, thus the user is implicitly expressing their interest in custom bikes, an advertiser on that site can ask the user's browser to collect and retain some information that might be used in the future if an ad from that advertiser will be displayed. Okay, now, note, importantly, that the advertiser learns exactly nothing about the visitor to the site. All of the information flow is into the user's browser, and only because of the website they're visiting.

Okay. Now, Google and I continue, I because I had to fix this language to even give us a hope of understanding it. So I clarified this. So they said: "Later, when the user visits a site with available ad space, the ad space seller, either a seller-side provider or the site itself, can use the Protected Audience API to run a browser-side ad auction which will select the most appropriate ads to display to the user. The ad space seller calls the browser's new - there's a function, navigator.runAdAuction() function, to provide the browser with a list of interest group owners who are invited to bid.

"Bids can only be provided by interest groups that the browser already became a member of when it had previously visited a website where it was able to collect that group, and when the owners of those interest groups had been invited to bid. Bidding code is retrieved from a URL provided in the interest group's configuration that was received earlier. This code, which is JavaScript, provides data about the interest group and information from the ad seller, along with contextual data about the page and from the browser.

"Each interest group providing a bid is known as a buyer. When the visited site's JavaScript calls the new browser function to run the ad auction, each buyer's bidding code generates a bid with the help of real-time data provided by their Protected Audience Key/Value service," whatever that is. "Then the advertising space seller receives these bids, as well as seller-owned real-time data, and scores each bid. The bid with the highest score wins the auction. The winning ad is displayed in a fenced frame" - which is one of those new APIs - "which absolutely prevents it from having any interaction with anything else anywhere.

"The ad creative's URL is specified in the bid, and the origin must match one in the list provided by the interest group's configuration, that same information that was received earlier. Finally, the advertising space seller can report the auction outcome, with a function known as reportResult(), and buyers can report their auction wins with a new function, reportWin()."

Okay. And finally, a bit later, Google offers a bit more detail, writing: "In the Protected Audience API, an ad auction is a collection of small JavaScript programs the browser runs on the user's device to choose an ad. To preserve privacy, all ad auction code from the seller and buyers is run in isolated JavaScript worklets that cannot talk to the outside world. A seller, a publisher or a supply-side platform, initiates a Protected Audience ad auction on a site that sells ad space, such as a news site. The seller chooses buyers to participate in the auction, indicates what space is for sale, and provides additional criteria for the ad. Each buyer is the owner of an interest group.

"The seller provides the browser with code to score bids, which includes each bid's value, the ad creative URL, and other data returned from each buyer. During the auction, bidding code from buyers and bid-scoring code from the seller can receive data from their Key/Value services. Once an ad is chosen and displayed, in a fenced frame to preserve privacy, the seller and the winning buyer can report the auction result."

Okay. Now, if all of this sounds insanely complex, you're right. This is not your grandpa's third-party cookies anymore. Nor are our web browsers simple apps running on our chosen OS to display HTML code. Those are the days that are long gone, and they're not coming back. It should now be abundantly clear to everyone that what Google has done with this Privacy Sandbox is to radically transform our web browsers from passive displays of whatever page is sent to them, into proactive advertising management engines. All of this new technology is already built into Chrome and has been there for the past six months.

Does all this probably give Sir Timothy John Berners-Lee, the web's original inventor, a huge headache? I would not be at all surprised if it did. Nothing less than an incredible

mess is required to deliver interest-driven advertising to users without revealing anything about those users to their advertisers. And by the way, "An Incredible Mess," as I said earlier, was the runner-up title for today's podcast. A large part of what I want to convey here is that nothing short of this level of complexity is required to protect our privacy while providing what the websites we depend upon, and want unpaid access to, say they need.

Now, the nature of inertia means that we would never, and I really mean never, move from the absolute mess we're in today to this new promised land were it not for a behemoth like Google to, first, carefully design and craft this solution, doing so openly and in full public view, inviting collaboration and industry participation at every step of the way, as they have; and, secondly, to then literally force it down the closed, choking throats of the rest of the existing advertising technology industry by taking Chrome, their world domineering browser, and gradually deprecating and foreclosing upon the operation of all of the previous tricks and techniques that have historically been used for user tracking and compromising users' privacy in the service of advertising tech.

No one else could do this but Google. This is not something where consensus could ever have been reached. It would never happen. It would be "committee deadlock." I've looked at the various ad tech blogs, and they're all screaming and pulling their hair out over this. But they're all also busily conducting experiments and getting ready for what they, too, understand is already inevitable.

Notice that one of the things Google has done with this reconceptualization of web advertising is to move the advertising auctioning process away from the advertiser and into the browser. Traditionally, an advertiser would purchase real estate on the web on website pages. Then they would run their own real-time auctions to determine which of their many advertising clients' ads should be inserted into that space for any given visitor, given everything that the advertising seller knows about the visitor from tracking them across the Internet. This changes all of that.

Now, all of the work is being done on the client side rather than on the server end, and doing this starves advertisers of all the data they were previously collecting while convincingly arguing against their having any further need to ever collect anything. In this new world, advertisers place static purchase offers to display content on website real estate with whatever ads they have to display, organized by interest group.

Using Google's new APIs, browsers that had previously visited websites representing various interest groups are now able to collect the advertiser's material that will later be needed to display ads for those interested. Then later, when browsers visit other websites with sell offers behind available advertising real estate, all of the information about the offers flows into the browser, which then itself conducts the auction and selects the ad that is most relevant to its user, based upon the places the browser has visited during the past few weeks.

The results of the auction are returned to all interested parties, and the ad tech company pays a piece of the action, or of the auction, to the site that offered up the real estate. In something of a follow-up, Google explains: "Understanding user interests can enable more relevant ads than just choosing ads based on site content (contextual targeting) or by using information provided by a user to the site on which the ad appears (first-party ad targeting). Traditionally, ad platforms have learned about user interests by tracking their behavior across sites. Browsers need a way to enable ad platforms to select relevant ads, so content publishers can get ad revenue without cross-site tracking. The Protected Audience API aims to move the web platform closer to a state where the user's browser on their device, not the advertiser or ad tech platforms, holds the information about what that person is interested in."

And that states it perfectly, I think. The way the entire web advertising world has worked until now is that every advertiser had to collect all of the information they possibly could about every individual who was surfing the Internet for the sole purpose of selecting the best advertisement to show them. The result was massively intrusive, massively redundant, and an ultimately ineffective utilization of resources.

But in the new world of Google's Privacy Sandbox, it's the user's browser that collects the information about its own user's interests by watching them navigate the web. As the browser moves around the web, future advertising opportunities are collected by the browser. And later, when visiting a site that is offering some available advertising space, the browser itself runs an auction on the fly to decide which of the opportunities it previously collected should be presented to its user based upon the criteria that it solely maintains.

This is obviously a big deal. But what seems just as obvious is that no lesser of a deal would get this important job done right. We can argue, and we'll always be able to argue - we certainly know that the EFF will always argue - that all website user-driven advertising customization should simply be ended, and that advertisers should settle for contextual advertising - placing their ads on sites which are offering content that's relevant and related to their ads - just like in the pre-tracking days. Unfortunately, multiple studies have shown that this would reduce website advertising revenue by about half, and many websites are barely making ends meet as it is. So the EFF's ivory tower stance is simply not practical, and it's never going to happen.

The only way to permanently end tracking is for it to be flatly outlawed. But tracking will never be outlawed while the case can be made that advertising customization is the only thing that's keeping today's web alive and financed, and that there's no alternative to tracking and compiling interest-profiling dossiers on everyone using the Internet. So what Google has done is to create a practical and functioning alternative. Tracking is no longer necessary. User privacy is preserved. And once this new system has been established, we can anticipate that we will finally see legislation from major governments - probably with Europe taking the lead - which will flatly and without exception outlaw any and all Internet user profiling and history aggregation because it will no longer be required.

Google's Privacy Sandbox masterpiece has been in place, as I've said several times, for the past six months. And although they've already been kicking and screaming, all other serious advertisers have been exploring it in anticipation of the future, which appears to be all but certain. As we move into 2024, fingerprinting will become increasingly fuzzy, and Chrome's third-party cookie support will be gradually withdrawn from its ubiquitous web browser. And finally, once the dust settles on all this, we can anticipate the end of the annoying cookie permission request pop ups.

Leo: I hope you're right.

Steve: We are heading toward a brand new web.

Leo: Do you think that, like Manifest V3, this will be adopted by other browsers at some point? Although as you pointed out earlier, Google has complete dominance in the browser usage.

Steve: They have complete dominance. Not only them, but all Chromium. So really it's Safari and Firefox that are the remaining wildcards. And this, I mean, this is what Google

is going to do. I think they've nailed it. You know, they have a solution. I mean, and the way they've nailed it is by massively burdening the browser with, like...

Leo: Well, I'm going to say that, is that my system is now working really hard to deliver ads. It just makes - by the way, the good news is this will be very easy to block.

Steve: Yes. And in fact, you can opt out of this.

Leo: Oh, can you? Oh, interesting.

Steve: Absolutely. There is a user-facing API that lets you just say no.

Leo: Oh. Okay. That's smart.

Steve: Google knows most people will not say no.

Leo: Right.

Steve: And I will not say no. If my use of the web is now private, and my browser is selecting the best ads for me to see, which is returning the highest amount of revenue to the websites I'm visiting, it is a win-win-win.

Leo: It's really an interesting idea. It's a great solution in terms of, you know, protecting your privacy, for sure.

Steve: Yes. It turned the entire model on its head. And the fact is today's, I mean, you know, once upon a time a browser was a little HTML rendering engine.

Leo: Yeah.

Steve: You know? Now it is literally a behemoth. I mean...

Leo: Well, that's one of the things that bothers me is now, I mean, the browser's going to be 90% of your CPU pretty soon.

Steve: It will. I mean, although it is little lightweight scripts. And we know that Google has a frenzy about performance.

Leo: Right, right.

Steve: You know. And how quickly this all displays.

Leo: Here's where Tim Berners-Lee might actually like this. He's been working toward a solution where you control your own data. You know, that your data is yours, and you lease it out, in effect, to people, which this is basically an implementation of. So it fits right into what Tim Berners-Lee has been doing of late.

Steve: Yeah.

Leo: So I think that it's possible Google may have found a way to give what we would like. Our holy grail would be for us to control our own information about ourselves, and then have the opportunity, if we wished, to share it, but at a price, you know, that we get something out of it. And that this is a step toward that.

Steve: As far as I know, there is no sharing opportunity. What there is in the UI is you can even browse the interest groups if your browser has said yes.

Leo: Yeah, yeah, see what you're saying, yeah, what it's saying.

Steve: And if you object to any, you're able to delete them, and you're able to mark them as never come back if you really don't want it.

Leo: Yeah. It's interesting.

Steve: No, they've really - they've nailed this.

Leo: Yeah.

Steve: I mean, and this is where they're going, and we know who "they" are. So, and their browser, it's funny, too, because I, you know, we've given a lot of space to the notion of fingerprinting, I think because it's kind of a cool technology. Everybody is still using cookies. Cookies is - and so when Google talks about right now, as of the beginning of the year, 1% of their users have third-party cookies turned off. And they're going to be, you know, they're doing that as an initial experiment. And then they're going to be deprecating the rest of third-party cookies. There will be no more third-party cookies by the middle of this year.

Leo: That's huge. That's so good, yeah.

Steve: And it is - and now so that's what's got the advertisers screaming and thinking, well, you know, we liked knowing all this about people. But we're going to have to fall in line.

Leo: This is the future, yeah.

Steve: It is the future.

Leo: And it really is a response to widespread ad blocking, the cookies, and other GDPR requirements. Yeah, I think it's interesting. Let's see what happens. They've thrown so many ideas up against the wall. None of them have stuck. This might be the one.

Steve: It does solve the problem. I see nothing wrong with it.

Leo: Yeah. Good. Thank you for filling us in. The Protected Audience API. Terrible name, but a very interesting concept, yeah. Your browser is the one that determines what you see.

Steve: Yeah. And even Privacy Sandbox. I mean, that doesn't tell you anything.

Leo: No.

Steve: Like, you know, don't kick sand in my eyes.

Leo: Good. We'll talk about it tomorrow with Jeff.