Brave's Brilliant Off the Record Request

Description: This week, before we address what I think is a brilliant new idea from the Brave Browser's Privacy Team, we're going to see why people are suggesting that the initials HP stand for "Huge Pile." What was Google thinking when they created the .zip TLD that no one was asking for? How has the Python Foundation responded to attacks and subpoenas? Do we believe a VPN service when it promises that no logs are saved anywhere? Will Twitter be leaving the EU? Does Bitwarden now support Passkeys? Who just got fined 1.2 billion euros, and why so little? What feature did WhatsApp just add? And what's the story about Google's new bug bounty for their Android apps? Then, after answering those questions, and a brief bit of good news about SpinRite, we're going to look at Brave's Brilliant "Off the Record" request concept and new feature.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-925.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-925-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here with lots of great things to talk about: why the .zip TLD is a really terrible idea; why Meta got fined $1.2 billion by the Irish and what they're going to do about it. And then Steve's going to applaud a really nice move from Brave to improve your privacy. It's all coming up next in Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 925, recorded Tuesday, May 30th, 2023: Brave's Brilliant Off the Record Request.

It's time for Security Now!, the show where we cover the latest news in the security sphere. And that's thanks to this guy right here, Mr. Live Long and Prosper, Steve Gibson. Hi, Steve.

Steve Gibson: And you know, Leo, we can make a whole meal out of today's Picture of the Week.

Leo: I haven't seen it. You told me, "Do not look at it until we're live." So I have not looked.

Steve: Only because I want our audience to be able to experience you seeing this for the first time.

Leo: Oh, dear.

Steve: So that they will be incented to get it themselves, to track it down in the show notes because this is the most inspired thing, I mean, I didn't think anything was going to beat the green ground wire stuck into the pail of dirt.

Leo: I did enjoy that, yes. I did enjoy that.

Steve: Yeah. This one, this is cleverness to a whole new level.

Leo: Is it a visual pun again?

Steve: It is. It'll take you a minute to visually parse it. You'll look at it, and you'll go, what? And then it's like, OMG.

Leo: Okay, I can't wait.

Steve: So today, this is Podcast 925 - I will next see you in June; this is our last podcast of May - titled "Brave's Brilliant Off the Record Request. Rarely do we come across something that is a simple idea that's new and really offers value. And the Brave - Brave as in the Brave browser. Their privacy team came up with something that I am really hoping that the other browsers adopt, and this becomes an industry standard. So we'll have a lot of fun talking about that. But first we're going to see why people are now suggesting that the initials "HP" stand for Huge Pile. We're going to ask what Google was thinking when they created the .zip top-level domain, which nobody asked for.

Leo: Nobody asked for.

Steve: Like nobody said, oh, gee, that's what we really need. How has the Python Foundation responded to attacks and subpoenas? Do we believe a VPN service when it promises that no logs will be saved anywhere? Will Twitter be leaving the EU? Does Bitwarden now support Passkeys? Who just got fined $1.2 billion euros, and why so little? What feature did WhatsApp just add? And what's the story about Google's new bug bounty for their Android apps?

Then, after answering all those questions, and sharing a brief bit of good news about SpinRite, we're going to look at Brave's brilliant Off the Record request concept which is a new feature that I hope, as I said, that the industry will adopt. And after you tell us about our first sponsor, Leo, you're going to be able to pull back the curtain on probably one of the best Pictures of the Week this podcast in its, I don't remember when we began the Pictures of the Week, but it's been decades. This is a topper.

Leo: That's saying a lot. The best ever. All right. I am ready. I'll tell you what. Let's do this all together. Everybody, oh, I've got to switch this over. All together now, we're going to look at the Picture of the Week. I might have to pull it up over here. Oh, no, there it is. Okay, you ready? We're going to look together. I'm going to scroll - oh, wait a minute. Do you want to see me as I scroll up?

Steve: No, we'll be hearing you.

Leo: You'll hear me? Okay.

Steve: It's pretty good.

Leo: We're going to watch together as we scroll up "The old-school way to add remote control [laughing] to a light switch." Well, well, well. I bet it works. Even when the power's out it works.

Steve: So if you don't want Russia or China to have control of the power...

Leo: If you're on the Battlestar Galactica, and you don't want the Cylons to get into your grid.

Steve: I think if we were to rewire the U.S. electrical grid using this technology, then there wouldn't be any concern about, like, IoT and SCADA attacks and so forth.

Leo: So tell us how this works.

Steve: It's just brilliant. So this would be if our grandfathers were inventive, and the switch that they needed to control was in the other room. It was like on the wrong side of the wall. So, huh, what do you do? So you drill two holes through the wall about six inches apart, one so that it comes out above the light switch and the other one so it comes below the light switch.

Leo: Yeah, yeah.

Steve: Then you run some half-inch PVC tubing through and put elbows on the ends in order to bend it down from the top and bend it up from the bottom.

Leo: It's a little over-engineered, but it gets the job done. Okay.

Steve: Then you thread a piece of string all the way through this contraption, drill a hole through the bat toggle of the light switch, tie the string to it, and then the string continues on down and back through the wall. So on the operator side of the wall...

Leo: It's two strings.

Steve: You've got two strings, yeah. And you pull the upper string, and it goes through the wall, pulls the switch up, and you notice while you're pulling the upper string that the lower string kind of goes in a little bit.

Leo: Yeah.

Steve: So it's like, okay. And now the lights are on, apparently in the room where you are, right, because apparently - it must be that this switch is controlling the wrong room. So someone said, huh. What are we going to do about that? You know, we want to control the lights in our room.

Leo: Hysterical.

Steve: So anyway, yeah, this is just...

Leo: Oh, my god, Steve, you've got a winner. You've got a winner.

Steve: Well, we have to thank - one of our listeners saw this and thought, okay, this is definitely going to make it for Picture of the Week. And they were correct.

Okay. So I had never heard it suggested that HP stood for "Huge Pile" until I went to catch up on the current state of those HP OfficeJet Pro 9020e series inkjet printers, as we know, all of which were effectively destroyed at the firmware level more than now three weeks ago. That was on the morning of Monday, May 8th, so three weeks and a day, right, 22 days ago. As a consequence of an aberrant autonomous firmware download and install. And to add insult to injury, it may be that the entire purpose for this badly failed update was simply to tighten HP's grip over the use of non-HP printer ink. But whatever the motivation, the entire world, which has now been waiting, as I said, for 22 days, is filled with HP-branded dead printers which no longer print - using anyone's ink.

And what's becoming more and more glaring with each passing day, now at 22, is that, incredibly, there has been very unsatisfying communication from HP about this issue. Any call to HP Service replies that HP is aware of the issue and, fear not, is working diligently on a solution. But we're talking three weeks now, and a day. So no matter what solution is eventually forthcoming, this has not been very impressive responsibility-taking on HP's part. I fully expected that when I checked back for today's podcast I would get news of, oh, here's what we're going to do. Static. Silence. Nothing has been said, which is just - it's astonishing that HP says, oh, yeah, go to whatever it is, support.hp.com and join the crowd. Wow.

Okay. So not quite four weeks ago, back on May 3rd, Google, which is now a fully fledged domain name registrar, announced their eight new top-level domains with a posting that was headlined "Eight new top-level domains for dads, grads and techies."

So the TL;DR on this is "Google Registry launches eight new top-level domains: .dad, .phd, .prof, .esq [short for esquire], .foo, .zip, .mov, and .nexus." For the dads they said: "Knock, knock. Who's there? With Father's Day right around the corner, .dad is here for the jokes, the games, and the advice. Whether you're a fit.dad, a gay.dad, or a dude.dad, .dad is the place to celebrate fatherhood." Leo, I think maybe they've gotten a little carried away over there in the Google domain registration world.

Anyway, they said: "Check out these interesting .dad websites." We've got Classic.dad, where you get to play a fun eight-bit game where your goal is to successfully mow the lawn while dodging pets and obstacles, and preventing weeds from spreading. Or Dear.dad. They say: "Check out this media platform dedicated to telling stories of Black fathers." Or Daily.dad. It's a new book from Ryan Holiday that provides 366 accessible -

you know, in case it's leap year - meditations on parenthood, a manageable slice for each day.

Okay. So for the grads they're saying: "May means graduation season for many in higher education. We're celebrating graduates and the professors who taught them well by launching .prof, .phd, and .esq. Whether sharing legal advice for everyday life or teaching courses on behavioral science, these new domains are perfect for showing off your credentials. Hats off to these early adopters," says Google.

And so we have Erika.esq. And they tell us that "Erika Kullberg is an attorney and money expert who is passionate about better positioning people for success."

Leo: Lot of attorneys would use esq. That would be popular, yeah.

Steve: Right. And so I think this brings a whole new meaning to the notion of vanity domains.

Leo: Well, exactly. When I started using weird domains, in other words, anything but .com, .net, and .org, people would always say, like I had Leo.fm. They'd say, oh, you mean Leo.fm.com. And I said no. And then you'd also see sites that would reject emails because they said, well, there is no .email domain. Yeah, there is, yeah.

Steve: Right. So we also have Casey.prof.

Leo: Which is good, for professors. That's fine.

Steve: Yeah. And that's Professor Casey Fiesler.

Leo: Or proofreaders would be good for. Maybe Elaine would like that.

Steve: Well, she would complain about the lack of two o's. You really, you know, proof with one O?

Leo: That's not good.

Steve: So Casey Fiesler is a technology ethics educator and science communicator, who apparently thought, hey, I need my own prof domain. And then we have Rafael.phd. Rafael is an expert in post-quantum cryptography. Fully homomorphic...

Leo: Well, he deserves his Ph.D.

Steve: Yeah.

Leo: Holy cow.

Steve: Well, also if you understand how fully homomorphic encryption works, you're a doctor.

Leo: Yeah.

Steve: And also privacy-enhancing technologies, and the application of these constructions. So, okay. But the real worry, and the reason for my bringing this up today is the new TLDs that Google chose to create for techies, that third category. Under the heading of Techies, they write: "May is also the month of Google I/O [as we know now], our annual developer conference. Whether you're learning to code, deploying a helpful tool, building your portfolio, or starting a new community, .foo, .zip, .mov, and .nexus have you covered. Here are some examples from our developer community." So we've got gamers.nexus.

Leo: Which is their actual name. That's the name of the company, the publication, yeah.

Steve: Right. Yeah, they said: "Use gamers.nexus to review computer hardware and plan your next gaming PC." Or "helloworld.foo: Learn how to code 'hello world' in...

Leo: You'd better explain why "foo" is for coding Hello World, though.

Steve: Leo, you tell us. You'll have fun with this.

Leo: All right. So there is an old military acronym FUBAR, effected up beyond all repair, I guess, I don't know. FUBAR. And so that's been around since World War II, right, along with SNAFU and a few other choice military - recognition. FUBAR is beyond all recognition. Okay.

Steve: Yes, recognition.

Leo: So programmers, when they were making up dummy variables in their texts, like Dennis Ritchie and Brian Kernighan and the C programming language, would use FU and BAR as...

Steve: Correct.

Leo: ...phony dummy variable names. But my question is, there's a .foo. Where is the .bar?

Steve: Yeah.

Leo: Foo without bar is like a...

Steve: You would think they would want to raise the bar, Leo.

Leo: .Foo is fun. I don't know who's going to use .foo. I might. Maybe should I go get Leo.foo? Maybe I will.

Steve: Yeah, you don't have enough.

Leo: I pity the foo.

Steve: And url.zip, they say.

Leo: Uh-oh. Uh-oh.

Steve: So there is apparently a domain, url.zip.

Leo: Uh-oh.

Steve: They said: "Create short, powerful, and trackable links with url.zip." So I guess somebody created, registered the domain and created a URL shortener called url.zip. And then, finally, david.mov, which is "Watch videos by David Imel in this liminal space."

Leo: Well, that's a $5 word.

Steve: Oh, boy. So, yeah.

Leo: Okay.

Steve: And just to finish off their posting, before we look at not only what could possibly go wrong, but what did go immediately go wrong, catastrophically, Google finished, saying: "Starting today" - this was like May 3rd; right? "Starting today, you can register all of these new extensions as part of our Early Access Program for an additional one-time fee. This fee decreases according to a daily schedule through the end of May 10." So the 3rd to the 10th, so that's seven days. So presumably on day one you had to really want one of these in order to belly up to the bar and pay whatever they were asking, which dropped down successively on each day until it was back to the normal price on May 10th.

They said: "All of these domains will be publicly available at a base annual price through your registrar of choice." Although I went over to Hover, and they don't offer ZIP, so they said, "Huh? What? No."

"To make it super easy for anyone to get their website live, we've worked with Google Sites to launch new templates for graduates, professors, and parents." Oh, good, you

don't even have to do any, like, website coding, just drop a template in. "To learn more about pricing or our participating partners, visit registry.google."

Okay. So it occurred to me that, if those grads are looking for jobs in Internet security, protecting the assets belonging to those dads, then Google's difficult-to-understand decision to offer domains under the .zip top level domain may create the full employment guarantee that those new grads have been looking for.

The malicious exploitation of these .zip TLDs took exactly zero time since their abuse was so obvious to everyone except for, apparently, Google. The new phishing technique is already going by the name "file archiver in the browser," as just one of many examples. It can be leveraged to "emulate" file archiver software in a web browser when a victim visits a .zip domain.

"mr.d0x" is a security researcher we haven't quoted in some time, although we did some time ago. He recently said: "With this phishing attack, you simulate file archiver software, for example, WinRAR, in the browser and use a .zip domain to make it appear more legitimate." In other words, threat actors are creating realistic-looking phishing landing pages using modern web tools - HTML, CSS, JavaScript - which mimics legitimate file archive software, hosted on a .zip domain, to elevate social engineering campaigns. The point is when you click on a zip file, and most people don't know if it's a single-click or a double-click, so they go with two because, you know, that's better than one.

So whether you single-click or double-click, if it's a URL, you're going to be taken to a site ending in zip, and the site is going to pop up and look like an archiving program. So you think you've opened the zip file on your local machine, when in fact you've opened your browser mimicking an archiver, and now you're under the control of this site when you click anything else that you want to use the archiving software for.

So the search bar in the Windows File Explorer will also emerge as a sneaky conduit where searching for a nonexistent .zip file opens it directly in the web browser should the file name correspond to a legitimate .zip domain. Our Mr.d0x said: "This is perfect for this scenario since the user would be expecting to see a zip file. Once the user performs this, it will auto-launch the .zip domain which has the file archiver template and is able to appear legitimate."

The problem, of course, that created all this is that the new .zip and .mov top level domains are also both legitimate file name extensions. So this invites confusion when unsuspecting users mistakenly visit a malicious website when they believe they've opened a file. They could then be misled into downloading malware.

Trend Micro agrees that this is anything but a good idea. They said: "Zip files are often used as part of the initial stage of an attack chain, typically being downloaded after a user accesses a malicious URL or opens an email attachment. Beyond ZIP archives being used as a payload, it's also likely that malicious actors will use zip-related URLs for downloading malware. The crux of the concern," they said, "is that with the introduction of TLDs that are identical to well-known file extensions, bad guys are going to cook up clever new ways to take advantage of this resulting confusion and ambiguity. If nothing else, it clearly equips bad actors with another new vector for phishing."

Then I got a kick out of Malwarebytes Labs. I mean, the tech security industry just went nuts over the idea that we now have a .zip TLD because it was so clear to everyone how bad this was going to be. Malwarebytes Labs, which titled their recent discussion of this ill-advised move by Google "Zip domains, a bad idea nobody asked for." They wrote: "If you heard a strange and unfamiliar creaking noise on May 3rd, it may have been the simultaneous rolling of a million eyeballs. The synchronized ocular rotation was the less

than warm welcome that parts of the IT and security industries," he said, "this author included, gave to Google's decision to put .zip domains on sale."

Okay. So then the author had some additional useful things to say which I think is worth sharing. He said: "Domain names and filenames are not the same thing, not even close; but both of them play an important role in modern cyberattacks, and correctly identifying them has formed part of lots of basic security advice for a long, long time.

"The TLD is supposed to act as a sort of indicator for the type of site you're visiting. .Com was supposed to indicate that a site was commercial, and .org was originally meant for non-profit organizations. Despite the fact that both .com and .org have been around since 1985," he says, "it's my experience that most people are oblivious to this idea. Against that indifference, it seems laughable that .zip will ever come to indicate that a site is "zippy" or fast, as Google intends." And it's like, so that's what zip - it's like, oh, it's because it's zippy. Okay. Quoting Google: "When you're offering services where speed is of the essence, a .zip URL lets your audience know that you're fast, efficient, and ready to move."

Leo: Oh, please. Well, but anybody can do it.

Steve: That's literally what they said, yes. There's no - you don't have to pass a speed test, Leo, in order to get the zippy .zip URL.

Leo: That's just a moron copywriter who wrote that; right? Just some marketing idiot. Because obviously it's dopey.

Steve: Yes, it is.

Leo: But I wonder what the technical thinking is. Maybe, whoa, let's make some money.

Steve: I just have no idea. He says: "Meanwhile, plenty of users already have a clear idea that .zip means something completely different. Since the very beginning, files on Windows computers have used an icon, and a filename ending in a dot followed by three letters to indicate what kind of file you're dealing with. If the three letters after the dot spell z-i-p, then that indicates an archive full of compressed zipped-up - files. The icon," he says, "even includes a picture of a zipper on it because reinforcement is good, and confusion is bad." Unfortunately, Google missed that message.

"As it happens," he says, "cybercriminals love .zip files and the last couple of years has seen an explosion in their use as malicious email attachments. Typically, the zip file is first in a sequence of files known as an 'attack chain.' In a short chain, the zip file might simply contain something bad. In a longer chain it might contain something that links to something bad, or contain something that contains something that links to something bad, or contain something that links to something that contains something that links to something bad. Anyway, you get the idea," he says. "The key to it all is misdirection. The attack chain is there to confuse and mislead users and security software."

Criminals use other forms of misdirection in file extensions, too. An old favorite is giving malicious files two extensions, which of course we spoke about at the beginning of this podcast 18-plus years ago, like evil.zip.exe. The first one, .zip in this case, is there to

fool you. The second is the real one, a dangerous .exe executable file type. Given the choice of two, users have to decide which one to believe. Most aren't even faced with that choice because Windows helps with the subterfuge by hiding the second file extension, the .exe, the one you really should be paying attention to, by default. So you just see something or other .zip.

"Domain names get the same treatment. Criminals make extensive use of open redirects, for example, web pages that will redirect you anywhere you want to go, to make it look as if their malicious URLs are actually links to Google, Twitter, or other respectable sites. Less sophisticated criminals just throw words like 'PayPal,' or anything else you might recognize, into the beginning of the link and hope you'll notice that bit and ignore the rest." Because, again, a lot of links just look like gobbledy-gook. You kind of look at it and go, what the heck? Oh, there's PayPal, okay, good.

He says: "Against that backdrop, Google inexplicably decided to introduce something that will generate no useful revenue, but will give cyber crooks an entirely new form of file and domain name misdirection, to add to all the others we're still wrestling with today. So what could criminals do with this new toy?" He says: "There is no better example than that provided by security researcher Bobby Rauch, in his excellent article 'The Dangers of Google's .zip TLD.'" In it, Rauch challenges readers to identify which of the following two URLs is a malicious phish that drops evil.exe on their machine?

And I've got the two links in the show notes. They begin: https://github.com/kubernetes/kubernetes/archive/refs/tags/v1.27.1.zip. And then the exact same URL except that in front of the final v.1.27.1.zip there's an @ sign. And he says: "The answer is that the evil link is the second one. The top one opens a zip file called v1.27.1.zip which was downloaded from github.com domain. The second one would go to the domain v1.27.1.zip, which in this hypothetical example triggers the download of whatever that site wants to provide.

Leo: I did not know that about URLs. So the @ sign changes the parsing.

Steve: Yes. And it turns out those forward slashes are not actually forward slashes. There's a Unicode character that looks the same.

Leo: Oh. So this looks like it is rooted at GitHub.com.

Steve: Yup.

Leo: But in fact those are not forward slashes until we get to the...

Steve: Exactly. Exactly.

Leo: I see. That makes sense. So it isn't just the @ sign. It's you have to use some tricks.

Steve: Yes. Unfortunately, all the browsers display that Unicode as a forward slash, and it passes right through the browser. So here's the problem for this audience. I would be hard-pressed to look at that second URL and think that that's a problem. It looks

absolutely legitimate, https://github.com. That's all you need to know; right? Anything to the right of that is going to be somewhere underneath GitHub.com.

Leo: We've taught everybody and we've trained ourselves to look for that root TLD. That's where it's rooted. And now we can ignore the rest. But no.

Steve: Yup.

Leo: Wow.

Steve: So me speaking: In other words, the Internet just became a whole lot more dangerous, even for those of us who take some pride in understanding all this stuff, in knowing how to examine URLs, and knowing our way around. What chance do our friends and relatives have?

Anyway, I've placed a link to Bobby Rauch's full write-up which explains in detail how simple it is to craft .zip URLs that look convincingly like URLs for well-known sites that are offering .zip archives for download, whereas in fact you are going to a site that somebody registered under the .zip TLD. You know, it is really a problem. And I just can't understand it. In fact, I've skipped over a bit of the Malware Labs stuff. But he said: "It's also possible that .zip will simply die on the vine if enough companies choose to block it.

Leo: Just block it, yeah.

Steve: Yes. Last week, Citizen Lab's John Scott-Railton urged his nearly 200,000 Twitter followers to simply "block it all," saying: "The chance that new .zip and .mov domains mostly get used for malware attacks is 100%." So, he said: "It's for you and your organization to decide if you should block it, but I will point out that if you are going to, the best time to do it is now. Almost nobody is currently using it, and nobody is going to use it in the future if it's routinely being blocked.

Leo: Little tip of the hat, too, to Firefox because I'm doing this in Firefox. This is the forged slashes. And in Chrome this would take you to a bad website. In Firefox, it says, uh, what?

Steve: Oh.

Leo: I don't think you're going where you think you're going, buddy.

Steve: Nice.

Leo: So that's Firefox I guess has decided to ignore those, or not treat those Unicode characters as slashes.

Steve: Well, and I mean, the well-known author Robert Lemos, writing for Dark Reading, his coverage was titled "Google's .zip, .mov Domains Give Social Engineers a Shiny New Tool."

Leo: Oh, yeah.

Steve: Wired headlined "The Real Risks in Google's New .Zip and .Mov Domains." TechTarget: "The potential danger of the new Google .zip top-level domain." And Ycombinator: "The Dangers of Google's .zip TLD." So, you know, you just don't see that when new top level domains come out for, you know, like .pizza and things. Who cares?

Leo: No, who cares? One other problem is, and this is another thing people should do, at least on the Mac, and I think on many browsers, they're set by default to "open safe files." So, and zip is normally considered a safe file. So when Safari sees a zip file, if you haven't unchecked that box, it will try to open it. And so you can go to a .zip and get what looks like a .zip that tries to open, and it's malicious. So another thing everybody should do, and I don't - on Windows, I think just as a convenience, browsers just assume, well, a .zip is safe.

Steve: Windows now puts it in the tree. It's in the File Tree.

Leo: Oh, that's terrible.

Steve: I hate that.

Leo: Oh, my, my, my.

Steve: It is so - I actually, I have subfolders where I put my zips because I don't want them to be expanded in this tree, the regular directory tree. Again, Microsoft wanted to make it easy. So thank you very much. Probably some way to turn that off, but I just - ugh.

Leo: Certainly if you're on a Mac, disable "open safe files by default" because that's terrible.

Steve: So Leo, after our next break we're going to talk about PyPI and something interesting that just happened to them.

Leo: Oh, yeah, because we talked about them last week.

Steve: They were under attack, and they've responded.

Leo: Yeah, yeah. It wasn't just last week. We've talked about them for many, many weeks.

Steve: Yeah.

Leo: Constant problems with these supply chain attacks.

Steve: Last week I noted that the Python Software Foundation had been forced to take the unprecedented step of temporarily shutting down all new account and Python package creation due to an overwhelming automated attack on the world's leading Python software registry. In response to the attack, plans are now underway to require two-factor authentication for all PyPI accounts; and, even more interestingly, to reduce the instances where PyPI portal needs to store a user's IP address. Okay, well, we'll get to that second one in a second because that, I thought, what?

Anyway, in a major, but welcome and probably inevitable policy change, by the end of this year, all accounts that maintain a Python library on the PyPI portal must choose some two-factor authentication method or have their access to PyPI features limited. The Foundation says that the move comes to improve their supply chain security of the Python ecosystem. The requirement for two-factor authentication is intended to help block account takeovers, which has been a problem in the past, and also to raise the bar for those automated attacks like the ones that PyPI has just been subjected to over the past few months.

Two-factor authentication has been supported and available, but its use has been optional, and there's not been much uptake because people just haven't cared that much. So that's what's changing. Last year the Foundation offered free security keys to the maintainers of its Top 1% of all packages in a move to show the PyPI devs that two-factor authentication would not be a serious disruption that some for some reason believed it would be.

Okay. But it's the second piece of news that I thought was the most interesting. As I was reading the reports that the Foundation was making, well, they said they were making an effort to purge IP addresses from their server logs, replacing them with hashes of the IP addresses, or in some cases geographic data provided by Feedly, which is their CDN. I didn't understand what that had to do with anything. The Foundation said that the only systems that will see plaintext versions of a user's IP address will be their rate-limiting systems, and even those they will work on replacing.

Providing some additional detail, the Foundation explained that they would be salting their hashes because the IPv4 address space, being relatively small at only 32 bits, would allow for the creation of hash-reversing lookup tables. As we know, the normal practice with salted hashes is to pick a unique per-hash salt at random, mix that in as you're hashing the thing you want, and then store the salt alongside the hashed value. This would allow a later IP comparison by using the same salt against a candidate IP to see whether the hashes match when you do the same thing.

But in this case, that normal approach doesn't provide the level of protection that the Foundation requires due to the fact that the 32-bit IPv4 search space is still so small compared to the speed of today's GPUs that are able to do very high-speed hashing. So instead, the Foundation said that they're going to use a single secret salt, that is, not a public salt, a secret salt which will not be stored in the database, and that they would be taking steps to protect it against any leakage.

Okay. So that was all well and good, but it still left unanswered the question of why the Foundation was bothering to hash their database's IP addresses? You know, what's the

point? The answer that came was somewhat chilling. It's the same reason ExpressVPN, a sponsor of this network, goes to great lengths to protect their users' IP addresses.

Okay. Get a load of this: The news of this major engineering effort came two days after the Foundation revealed that it had been subpoenaed by the U.S. Department of Justice to reveal information on five PyPI accounts. The requested information included the users' IP addresses, but also the IP addresses of all users who had downloaded packages from those five users. The Foundation wants to eliminate their own ability to respond to such subpoenas in the future, thus protecting the IP addresses and thus to some degree the identities of those who are uploading and downloading Python libraries.

So the problem here is essentially identical to the issue with encryption. While no one wants to provide care and comfort to criminals, there's also a reasonable presumption and expectation of privacy. And unfortunately we keep seeing instances of government overreach, such as the other story we covered last week where the U.S. FBI abused the powers of the FISA data collection to improperly search the personal communications of Americans more than 300,000 times in a little over one year, according to the U.S. Senate Intel Committee. If the government wants the trust of its citizens, the government needs to first be worthy of that trust. These revelations, you know, about them not being worthy don't engender much trust. And now we have very strong privacy enforcing technologies that are going to move their ability to have the information that they want, even under subpoena, out of their reach.

Other package registries have also announced plans to require their users to enable two-factor authentication. And while the Python Foundation's move to purge IP addresses from logs and databases is a novel solution, it may develop into more standard practice in the future. So as a fly on the wall, I'm curious why the Department of Justice produced a subpoena that required that the Python Foundation comply. In their full-disclosure posting they said they were caught by surprise, they had the information, and they complied because at that point they had no choice. But they're going to move that information out of their own grip so that they are not able to comply in the future.

And speaking of not being able to comply in the future, an object lesson story appeared in the news last week. The so-called Super VPN service by the publisher SuperSoftTech, whose Android app - and there also is one from iOS. The Android app has been downloaded more than 100 million times, was exposing more than 360 million records due to its Chinese developer leaving the app's database open and exposed to the Internet. And that database contained all of the information the VPN provider explicitly promised would never be exposed, including information on all of its paid customers, including details such as emails, the customers' real physical IPs, geolocation information, and VPN servers they have connected to.

The database was secured only after it was discovered and reported to the developer by security researcher Jeremiah Fowler. And it's worth noting that Jeremiah's findings contradict the app's Google Play Store listing at the time - it's been since removed - where the service claimed "no logs saved anywhere" as one of the benefits and features of its use. Our obvious takeaway lesson here is that it's very easy for anyone to make the claim to not be retaining any logs. No one using a VPN wants their usage to be logged. There's zero benefit to the user from having their VPN provider logging their activity. But it's obvious that the anti-logging claim needs to be true for it to have any value whatsoever. So our takeaway is that it really does matter who is making the claim.

Well, Twitter may be facing stiffer headwinds with the EU's 27-nation bloc after Twitter last Friday chose to drop out of a voluntary European Union agreement to combat online disinformation. The Associated Press reported that European Commissioner Thierry Breton tweeted - actually he used Twitter, right, he tweeted - that Twitter had pulled out of the EU's disinformation "code of practice" that all other major social media platforms,

including Google, TikTok, Microsoft, Facebook, and Instagram, among others, have all pledged to support. He added that Twitter's "obligation" remained, referring to the EU's tough new digital rules taking effect in August. Breton said: "You can run, but you can't hide." Okay, well, so a little bit of tough talk there.

There were early signs that Twitter was not prepared or planning to live up to its commitments. The European Commission blasted Twitter earlier this year for failing to provide a full first report under the code, saying that it provided little specific information and no targeted data. Breton said that the new digital rules that incorporate the code of practice fighting disinformation will become a "legal obligation." Okay. But, you know, Elon as we know is notorious for ignoring any legal obligations that he doesn't actually need to heed.

Breton concluded by adding, somewhat ominously: "Our teams will be ready for enforcement." Uh-huh. It will be interesting to see what, if anything, that amounts to. Perhaps some monetary fines that Elon will ignore. I think that's going to be about it. I doubt that the EU has the huevos to block all Twitter access outright. So I think Elon is probably just going to be successfully calling their bluff.

There was also a flurry of confusion in the tech press last week over Bitwarden's announcement of their increased support for Passkeys. I read some misreporting, and I thought, wow, I mean, before I knew it was misreporting I thought, great, I can tell everybody that Bitwarden will now be able to operate on Passkey-based sites. No. The actual news was not what we've been hoping for, at least not yet. Apparently it's coming soon. Bitwarden is saying that end users will be getting Passkey support for their Bitwarden clients sometime later this summer.

What was the news was that their Passwordless.dev site, which we referred to at the time shortly after Bitwarden acquired that group that they acquired not long ago, that they are offering APIs and code libraries now formally available to enable enterprises to add backend Passkey WebAuthn support into their authentication flows. It's all open source, and it's going to make it much easier to do that.

And there was also news that Bitwarden client users would be able to use Passkeys to authenticate to their Bitwarden client, which makes the entire chain, as soon as the client is able to authenticate to websites, will make that whole authentication chain complete. So, you know, Bitwarden doing good stuff, making WebAuthn more accessible. But we don't yet have a client that we can use through Bitwarden in order to log into websites. And again, even if we did, where would we log in because there's no one yet really supporting it.

As I was scanning recent news for interesting updates and discussion, the phrase "1.2 billion euro GDPR fine" was difficult to ignore. It appears that Ireland's Data Protection Commission wanted to get everyone's attention. So they levied a 1.2 billion euro fine against Facebook's parent, Meta, for their failure to comply with the EU's GDPR laws. The Irish officials claim that Meta illegally transferred the personal information of EU users to the U.S. without their approval.

What's apparently meant by this is the issue of where Facebook data resides, as in, you know, in a data center, sometimes there, sometimes here. So this IDPC (Ireland Data Protection Commission) has ordered the company to cease all data transfers and delete existing user data within six months. And in case anyone was wondering, yes, 1.2 billion is indeed the largest fine so far imposed - I guess I would say requested? I don't know.

Leo: They're going to appeal it, obviously. It's going to be a while, yeah.

Steve: Yeah, under the EU's GDPR. Now, to my mind this represents a failure of imagination, Leo, on the part of those plucky Irishmen. If they wanted to really get everyone's attention over a fine that was never going to be paid anyway, they should have gone directly to 1.2 gazillion euros.

Leo: Well, they can go as high as, I think, 10% of revenue. They can go pretty high.

Steve: That's true.

Leo: Yeah. Not a gazillion. I think no one makes that much money.

Steve: Well, why settle for a mere billion when you could ask for a gazillion?

Leo: Well, here's my question. And we talked about this on Sunday. You know, email requires that my message goes from the server in my country to the server in your country. I presume Facebook Messages and Facebook posts have the same thing. It doesn't make sense in a modern Internet to say your data has to stay in our country. I understand why they would want that. But doesn't it break the Internet?

Steve: Meta, for their part, said: "Let us think about it. Okay, no."

Leo: That's probably the right answer, actually.

Steve: To exactly no one's surprise, Meta said that they would be appealing, as you said, Leo, this nonsense. In somewhat more detail, Meta explained that they were being unfairly singled out for attack because thousands of businesses, maybe even a bazillion businesses and organizations rely on the ability to transfer data between the EU and the U.S. in order to operate and provide their everyday services. They explained that the real issue was not about one misbehaving company's GDPR-violating privacy practices; but rather that there are fundamental conflicts of law between the rules in the U.S. and in the EU, and that it's those laws that need to be brought into alignment.

And, they said, notably, respective policymakers are expected to be resolving all of this in this coming summer. So yes, Meta formally stated that they would be appealing the ruling, including the unjustified and unnecessary fine, and seek a stay of the orders through the courts. And meanwhile, Facebook continues unimpeded throughout Europe. So, wow. Just, I mean, crazy. I guess, you know, if you have the GDPR, then you want to try to use it.

Leo: Yeah. I mean, I don't - I honor their desire to protect people's privacy. I understand that. And, you know, some fines, like the fine against Google for saying we turned off location tracking but didn't...

Steve: Yeah.

Leo: That doesn't bother me at all.

Steve: Yeah.

Leo: I just think that - I don't understand how the Internet's supposed to work if everything has to stay - it's like, well, your email will only work in Ireland. Like you can only email people in Ireland. Huh? You can only message people in Ireland. I don't understand what they're thinking.

Steve: Well, that's why it's called iMessage. It's Ireland Message.

Leo: Ireland Message.

Steve: That's right.

Leo: Apple knew this all along.

Steve: That's right. Okay. So while we're on the subject, very briefly, of Meta and Facebook, I wanted to note that WhatsApp messaging announced on Monday, actually Monday before last, not yesterday, that they would be adding a feature that iOS users received with the update to the "16" versions of iOS and iPad OS, which is the ability to edit after the fact, for up to 15 minutes, the content of any recently sent message. In the case of iOS, Apple allows up to five edits of a message during that 15-minute having been sent window, which I suppose is useful if you're really having a difficult time getting the message right.

Anyway, as it happened, I was glad to be reminded of this new feature in iOS 16 since I have a buddy who frequently follows up messages with little asterisks and like single words where he rereads the message after he sent it and goes whoops, and then lets me know that he realizes there was a typo. Anyway, since I put this in the show notes last night, I sent him a text and said, "Hey, guess what, Mark, you're able to edit your messages after you send them." So he's very excited. So just a reminder to everybody else who's using WhatsApp or iOS and iMessage and iPad OS, you can now edit things after you send them.

And finally, Google has announced a new bug bounty program aimed at their own franchise's Android apps. They've named it Mobile VRP for Mobile Vulnerability Reward Program. And in describing the program they said: "Google's Mobile Vulnerability Rewards Program (Mobile VRP) focuses on first-party Android applications developed or maintained by Google." And by that they mean "or Google property."

"The Mobile VRP recognizes the contributions and hard work of researchers who help Google improve the security posture of our first-party Android applications. The goal of the program is to mitigate vulnerabilities in first-party Android applications, and thus keep users and their data safe. Only apps published by the developers in the list below, or apps in the Tier 1 list, are in scope for the Mobile VRP." So that's apps by Google LLC, Developed with Google, Research at Google, Red Hot Labs, Google Samples, Fitbit LLC, Nest Labs Inc., Waymo LLC, and Waze (W-A-Z-E).

So there are three general classes of vulnerabilities that qualify under their rewards program. The first of the three is Arbitrary Code Execution where they explain: "Vulnerabilities of this type allow an attacker to execute arbitrary code in the context of

the vulnerable application. In order to qualify, the ACE (Arbitrary Code Execution) should allow an attacker to run native code of their choosing on a user's device without the user's knowledge or permission, in the same process as the affected app." Meaning there is no requirement that the OS sandbox needs to be bypassed.

And they provide three examples: an attacker gaining control of an application, meaning that code can be downloaded from the network and executed; or overwriting a .so file with a malicious .so file which is then executed by the victim app; or executing Java code in order to call the "exec" function which is then able to run arbitrary native code. They said that merely tricking a user into installing an app and executing code within that app itself does not qualify. So it's got to be your ability to execute your own arbitrary code within one of Google's family of apps.

The second of the three qualifying classes is the theft of sensitive data, which includes vulnerabilities that lead to unauthorized access to sensitive data from an app on an Android device, where the scope of the qualifying data is data that enables unauthorized access to a user's account, you know, login credentials, authentication tokens that are able to perform sensitive state-changing actions that result in non-trivial damage to the victim. Or sensitive user-generated data: contact list information; photos unless they've been made public by default; content of a user's messages (email, IMs, or texts); call/SMS logs; web history; or browser bookmarks. So basically getting data that you're not supposed to be able to get from any of Google's apps. And finally, information that is linked or linkable to an individual, such as medical, educational, financial, or payment data, including employment information.

So they did note that location information alone does not qualify. They don't feel that's significant enough. But all the other stuff that you should not have access to does. And lastly - those were the first two classes. The last one is just additional vulnerability types which are in scope: path traversal and zip path traversal vulnerabilities which lead to an arbitrary file write; intent redirections leading to launching non-exported application components; vulnerabilities caused by unsafe usage of pending intents; and orphaned permissions.

The range of rewards is as high as $30,000. You would need to do a zero-click arbitrary code execution in one of their apps. But if you can do that, you've got $30,000. And they go down to about 750 bucks, depending upon severity and class of app. So anyway, what we have witnessed is the emergence and clear success of the concept of bug bounties as like a functioning chunk of the security infrastructure in our industry where good guys are paid for finding and responsibly disclosing previously unknown and important bugs. And if you're good at it, you can make a living.

I wanted to just quickly note that since last week I finished the full rewrite of SpinRite's mass storage data operations backend, and I finished testing it all by the end of the weekend. So Sunday evening I released the 28th alpha, Alpha-28, to the gang over in GRC's spinrite.dev newsgroup, and they have begun putting it through its paces. I have not been looking obsessively at it yet because I've been working on this podcast. But it looks like it's going pretty well. You know, I'm sure I'll have something to report about that next week.

In my own testing, I can say that it felt significantly more solid and responsive than it had before, and it did incorporate many new features, essentially thanks to all of the revelations that I was experiencing over the last six months. So I'm very glad that I took that month to scrap and reengineer that very important part of SpinRite. That's where all the actual work happens.

Leo: All right, Steve. I've been kind of playing with the Brave browser. In fact, a couple of weeks ago Paul Thurrott made it his app of the week. Sounds like maybe it's a good browser to use for privacy.

Steve: Well, they've done something I think is very cool. Last Wednesday the Brave browser's privacy team announced a slick new forthcoming feature that as I said at the top of the show I would love to see obtain some traction within the wider browser community, meaning also being adopted by Google for Chromium and Mozilla for Firefox. It's called "Request OTR," where OTR is short for "Off the Record." The shortest summary of this interesting new idea is that this feature allows websites to suggest to the browser that this user's visit to the site should probably be forgotten and thus remain "off the record."

So here's how the Brave Privacy Team explained their idea. They said: "Starting in version 1.53, Brave will begin rolling out a new feature called "Request Off the Record." This feature aims to help people who need to hide their browsing behavior from others who have access to their computer or phone." Now, initially this sounds like incognito mode; right? But it's not.

So they said: "For example, a person who's the victim of intimate partner violence who needs to find support services without their partner knowing, or someone needing to find personal healthcare without others in their home finding out. Request OTR allows websites to optionally describe their own content as 'sensitive.' The browser can then ask the user if they would like to visit the site in OTR mode, where the site is visited in a clean, temporary storage area. Sites visited in OTR mode are not saved to your browsing history, and any cookies, permissions, or other site data do not persist to disk. Meanwhile, all other normal sites visited are stored and treated normally, hiding the fact from anyone who may access the device later that any 'unusual' behavior happened.

"Brave intends to work with other browser vendors to standardize OTR, so that at-risk browser users can be private and safe across the web, regardless of which browser they're using. This feature has been designed with the input of, and in collaboration with, several civil society and victim advocacy groups. We agree with Mallory Knodel, the CTO at the Center for Democracy and Technology, who said: 'Brave Browser's attention to detail with OTR Mode, where users can more easily choose which websites are recorded in their browsing history, is an important privacy innovation that can protect users in "attacker you know" situations or anyone who wants more control over what their browser remembers and what it doesn't. This feature empowers people who browse the web - all of us - and gives us more agency over content consumption.'"

Okay. So Brave said: "Some users need to hide their browsing from people who have access to their device. Most often, when people talk about web privacy, they're talking about protecting personal data from other websites, for example, blocking Google from recording the sites visited. However, web users have other privacy needs, too, needs that are currently poorly served by most browsers." They said: "Consider Sarah, a hypothetical web user who lives with Stan, a physically abusive partner. Sarah needs to use the web to learn about legal, medical, and other support services in her area so she can safely exit their relationship. Stan, though, suspects Sarah may be planning to leave, and begins monitoring Sarah's phone, computer, and other devices to see if she's contacting support services.

"Unfortunately, not only do browsers fail to protect users like Sarah, they actually make it easier for abusers like Stan to digitally surveil them. Browsers record a wealth of information about our browsing behavior and interests, both explicitly (browsing history, DOM storage, and cookies) and implicitly (cache state, saved credentials, URL

autocomplete). Worse still, the tools browsers do include to protect people like Sarah are incomplete and/or difficult to use correctly.

"Browsers currently provide some tools to help users hide their activity on sensitive sites. However, these tools are insufficient to protect people whose safety depends on hiding visits to specific sites from people who have access to their device. Existing tools either hide too much, thus inviting suspicion from abusers; too little, thus allowing abusers to recover browsing history; or are otherwise difficult to use successfully.

"Private, also known as incognito, windows allow users to browse the web without their browsing activity being permanently recorded. Unfortunately, private windows do a poor job protecting users from on-device surveillance. It's easy to forget to open a private window before visiting a site, especially under stress, thus causing the site visit to be permanently recorded. It's equally as easy to forget to close the private window, and thus continue browsing in the private window beyond just the target sensitive site. This can reveal to the abuser that private browsing modes have been used, which on its own may elicit suspicion or put the victim at further risk.

"Similarly, some browsers include advanced browser controls that could be used to delete browser storage for specific sites. This approach has the drawback of needing to be performed after the site was visited, instead of protecting the user during the visit, which may put the user at risk if the browser needs to be closed quickly. Additionally, these controls are often difficult to find, and more difficult still to use correctly by non-technical users. And finally, these browser controls typically only allow the user to delete stored values for the site, for example, cookies or permissions; but do not allow the user to delete other traces of the site, for example, browser history or caches.

"Finally, some sensitive sites include quick-exit buttons in the site themselves, which allow a visitor to quickly navigate away from the site in a way that may be semi-difficult for an abuser to detect. While useful, this approach is also incomplete. Quick-exit buttons cannot delete many types of site data, for example, permissions or caches; and are constrained in their ability to modify the browsing history. Further, they depend completely on the correct implementation by the site. The browser is unable to protect the user. In contrast, Brave's "Request OTR" approach provides a comprehensive way for sensitive sites to request to be omitted from a user's browsing history and local storage.

"Any site can request to go off the record, and the user is prompted to determine whether they would like to do so. If so, the Brave browser creates a temporary storage area for that site, and does not record the site visit in the user's browsing history. The OTR session is tied to the site, and any other sites the user visits, even in the same tab, along with any sites visited in any other tabs, are recorded in browsing history as usual.

"Brave's implementation of Request OTR protects the user in the following ways: The user is prompted and proactively asked upfront whether they wish to have their visit forgotten after they leave. The user is protected the entire time they're visiting a sensitive site. They don't need to attempt to scrub their browsing history later. Other, non-sensitive sites are recorded as usual, which prevents the appearance of large gaps in browsing history that might look suspicious to an abuser. All target site behaviors are prevented from persisting to disk, including cookies, caches, browsing history, permissions, et cetera. And finally, OTR prevents sites from abusing the feature. A site cannot go off the record unless a user explicitly gives the site permission to do so.

"Brave has developed Request OTR specifically to help people suffering from intimate partner violence, or people otherwise needing to hide visits to sensitive sites from their browsing history. However, OTR is intentionally a general browser feature and is intended to be usable by any site on the web. There are currently two ways for a site to request to go off the record in Brave.

"The primary, intended way is for the site to include the header Request-OTR: 1 in the website's response to the initial navigation request to a site. If the browser receives this header, the browser will halt the navigation and ask the user if they would like to visit the site off the record. If the user says yes, then the browser does two things. It does not record the site visit in the browser history, and it creates a temporary storage area for all caches, cookies, and permissions. The browser continues using this temporary storage area for all subsequent pages visited within the same tab, within the same site. When the user closes the tab or navigates away from the site, the temporary storage area is discarded, and browsing behaviors return to being recorded as usual.

"The second way a site can request to go off the record is to be included in Brave's preloaded list of 'request off the record' partner sites. These are sites that serve victims of intimate partner violence and have told Brave they're interested in being considered a sensitive site, by default, thanks to the browser, in the browser. This list is intended as a bridge measure until all such sites have the opportunity to implement the previously mentioned header approach.

"There are a few caveats: Users should be aware that Brave's Request Off the Record feature cannot protect users from other software on their computer that might record information about what sites they visit. Examples of software the Brave browser cannot hide browsing history from include browser extensions; network spying; malware or spyware installed on the device; information saved by sites before or after you visit off the record, in other words, that's obviously just out of scope; operation system level logging; and crash logs." They said: "Brave is exploring what additional protections can be provided using such threats, but users should be aware that as with systems like private browsing mode, Brave's Request Off the Record mode only prevents recording of core browsing behaviors and data."

They finished, saying: "We're excited to release Request Off the Record in the upcoming version 1.53 of our desktop browser, with an Android version coming in 1.54. We'll be rolling it out to users shortly, though people interested in testing the feature can now enable it by visiting brave://flags and enabling #brave-request-otr-tab." They said: "Please note this should only be done if you understand the risk of testing experimental browser features," blah blah blah.

"We're also excited about the next steps we're taking to further improve the Request OTR feature. First, we're working with experts and researchers at George Washington University and Paderborn University to evaluate how Request-OTR is understood by users, and how we can further convey to users exactly what protections the feature does and does not provide. We will both share the research that results from this collaboration on this blog, and incorporate it into future versions of Request-OTR in Brave."

And finally: "We're interested in working with other browsers, organizations, and web companies to potentially standardize Request-OTR so that users of other websites and browsers can benefit from the protection. Our current implementation is the result of working with a wide range of abuse advocates, technologists, browser specialists, and NGOs; and we're eager to continue working with similar organizations to best support web users."

So to all of that foregoing, I say a huge bravo. As I said, in 2023, with our web browsers at their current level of maturity, it is extremely rare to encounter something so clear, so clean, and so simple that is, I would argue, able to offer new and useful benefits to a user's browsing experience. And this new feature is such a thing. The user's experience of this is perfect. It could be baked into every browser, and most of us would never have any idea that it was present. But the moment someone who was in an environment that might make them vulnerable, visited a website that understood that by virtue of the services it offers, its visitors might need the site's help in protecting themselves, the

website could immediately prompt the browser to proactively ask the visitor whether they would like all of their use of this site to remain off the record and be immediately forgotten after they've left.

You know, it's clean and simple and I think a wonderful addition to the traditional incognito mode of browser usage. So I hope that the clarity and the objective goodness of this idea captures the imaginations of those at Google and Mozilla, as well, and that it would grow into a World Wide Web standard that I think it deserves to be. Just you know, simple, but, you know, and easy to do, but I think really a worthwhile benefit.

Leo: There's some question in the chat about how this differs from incognito mode or a private browsing mode.

Steve: I would say that it's that that requires you to take that action explicitly. It also requires that you not do other things. The problem is while you're in there, your browser's not logging.

Leo: Right.

Steve: So if somebody was suspicious and looked at your history, they would see a block of time when they knew you were on the computer and doing something that there was no record. Also, if you suddenly got interrupted, you have to close the incognito mode. You might forget and leave it open. So, and most of what I like about this is that somebody innocent who wasn't thinking in advance is proactively asked, oh, you're visiting us. Would you like this to be forgotten?

Leo: Right.

Steve: And they think, oh, yes, thank you.

Leo: Yes, yes.

Steve: And so it's that proactive - and again, no sites would ever display this to us unless the site itself knew that the type of services they were offering were those that they...

Leo: Right, like a women's health clinic or, yeah.

Steve: Yes.

Leo: Of course, yeah, yeah.

Steve: Yes.

Leo: I think that's great.

Steve: I think it's just brilliant. It is simple. It's a tiny little thing, trivial to do. And so what it is, is whereas incognito mode is mostly the things I'm doing I don't want Google to remember or my browser to remember, whatever it is, you know, any tabs that you open. This is just it's more selective just for the site that you're visiting. So other things you do during the time, other tabs you visit, other links you click, they'll all be there, just not this one selective site. And for me also, the fact that it's proactive is the key, that this type of site says, would you like this not to be remembered by your browser? Oh, yes, thank you. Of course you're also free to decline it. You say, no.

Leo: I don't care.

Steve: I'm fine, yeah. I just think it's very clean.

Leo: Yeah, very nice. Steve, always a fascinating and informative podcast. Thank you so much. Episode 925 is in the books.