Side-Channel Privacy Leakage

Description: This week Steve and Leo examine the many tiny bits of individually non-unique information that inherently leak from a user's web browser out on the Internet. What's surprising is that when all of these individual non-unique bits are gathered together and assembled into a single "fingerprint," the result IS often unique and can thereby be used as a tracking fingerprint to identify individual users' movements as they surf.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-264.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-264-lq.mp3

Leo Laporte: This is Security Now! with Steve Gibson, Episode 264, recorded August 31, 2010: Side-Channel Privacy Leaking.

It's time for Security Now!, the show that covers your privacy needs, protects you online, makes sure the bad guys aren't winning. And here he is, the good guy from Security Now!, our Chief Security Officer, Steve Gibson of GRC.com.

Steve Gibson: The good guy.

Leo: The good guy. The white shirt.

Steve: I'm on your side, yes.

Leo: Steve is, of course, a prolific software writer. He's got a lot of great tools on his website, GRC.com. And we do this podcast every week, now in our sixth year of covering security. So we're glad, if you're new to the show, we're glad you're here. And stand by. Stand by for revelations. So, Steve, what are we doing today?

Steve: Well, we've got a great topic. This follows off of the EFF's project, which they called Panopticlick. And it's part of a - it was sort of a research project that the EFF started at the beginning of 2010, which ran for about six months, and culminated in a paper which they submitted to a recent privacy conference. I've been sort of waiting for that to happen because I wanted the summary of what they learned from this

experiment. And the experiment was, independent of cookies and the problem we understand with cookies being used to track people, what other privacy consequences does surfing have? That is, are there other ways that identifiable or trackable data of any kind leaks from people's machines?

And in crypto we've talked, there is a term called a "side-channel attack," or "side-channel leakage." For example, you might have a hashing function which you're using to generate a signature, where you put data in, and out comes a hash. And that all seems very secure. Except little things, like the exact timing of how long that process takes, or how long aspects of it take, something you wouldn't even think of that's sort of completely off on the side, can be used to leak information.

And so the title for today's podcast is Side-Channel Privacy Leakage, which is things that are going on which, believe it or not, commercial companies now exist to exploit. There's robust tracking technology, independent of cookies, which works. And so we're going to talk about what those are, how good they are, what their nature is, what the EFF learned through its experiment, and what people can do about it.

Leo: Sounds great. I can't wait.

Steve: Good stuff. Bizarrely enough, nothing happened in the last week.

Leo: Wow. Nothing happened.

Steve: Nothing. No security news. No updates. What happened, as I was looking at the dates of these things, the big news with - there was a new Chrome, Google Chrome browser. There was an Apple update. There was an Adobe release. That all happened exactly on last week's podcast.

Leo: Yeah, a ton of stuff, yeah.

Steve: So we were able to cover it very freshly. Nothing since then. So we have, for the first time in a long time, no news and no updates. I did want to let our listeners know that I'm very pleased with this latest Kindle which has come out. I don't know, if we were numbering them, I think Kindle 2 would have been the successor to the little first wedge-y one. Then of course we had the DX, which was the big sort of trying to be a textbook, a 10-inch screen, which I still think you need color for that. And you really need to be able to scroll PDFs. So I don't know how the DX is doing.

But the Kindle, I guess maybe we're on the Kindle 3 now, is smaller than the Kindle 2. Same screen size, so they just reduced the margin. Its page turn, the speed at which pages can be turned is as fast as you can go with a book now. I mean, it's much faster than the Kindles have been before. And the price has dropped to $149 for a WiFi-only version, or $189 for the WiFi plus 3G. So, and in addition, Staples, many hundreds of outlets of Staples stationery and office supply stores are going to be carrying them sometime here in the fall. So I'm just - I love my latest Kindle 3; and I wanted to let our listeners know that that had happened, in case they were curious.

Leo: It's going to be a very significant product, I think, for Amazon because of the price. At $139, that's a very compelling price. And as you say, putting it in the stores makes it accessible to anybody to hold and to - and I just think that Amazon's responded exactly properly to the iPad challenge.

Steve: Yes, well, iPad, and of course we do have Barnes & Noble is still on their trail, also, trying to compete with their books. And I read that Barnes & Noble had 1.5 million titles. It's like, wait a minute, that seems like a larger number than I would expect.

Leo: They include the Gutenberg titles that they have, the public domain titles.

Steve: The ones no one really wants to read.

Leo: Well, but some people want, I mean, look, if you're going to buy Jules Verne's "Twenty Thousand Leagues Under the Sea," you're going to buy it, whether it's public domain or not. But that does add considerably to the number.

Steve: Good book. And I did have just a short - since we didn't have much news and everything, I didn't want to bog everybody down with a long SpinRite story. Just a little quickie that was sent through the Security Now! feedback page, that is, for this podcast. It started out, "YASSS!" which then he in parentheses says "(Yet Another SpinRite Success Story): My stepson Matt could not boot his Windows XP computer as it complained of a missing file. His future father-in-law lives near him and is something of a geek. He tried booting in repair mode for Windows, but that failed, too. After weeks of messing around, and even trying to install Linux, he called me. I sent him my SpinRite disk, and it solved the problem in a little under two hours. The only downside is I had just about talked him into installing Ubuntu Linux. Now that his XP is fixed, he may not follow through." And we have a little frowny face. And he says, "Love the show. Jon Payne, Atlanta, Georgia."

Leo: That's pretty funny. Keeping Windows alive everywhere.

Steve: At any expense.

Leo: At any cost.

Steve: At the cost of Linux.

Leo: All right, time to talk leakage.

Steve: Okay.

Leo: Seems like a personal problem, but go ahead.

Steve: So I ran across a euphemism for this that I really liked a lot: "nonconsensual user tracking."

Leo: Mm-hmm.

Steve: Nonconsensual.

Leo: Nonconsensual user tracking.

Steve: Now, we know that cookies have been a long-term problem for many people who sort of philosophically probably more than anything else object to the idea that in some fashion their actions, their movements, are being tracked across the Internet. The way this happens is sort of never - it was never intended. And we've covered this, so I won't go into it in too much detail, in the past. We've covered it in the past.

But the idea with a cookie was that it would be used for a single website to sort of uniquely tag its visitors so that, while they were there doing things, their individual queries for web pages would send back this tag, so that the website could - the term is "keep some state." It would know that they were - it would sort of know that this was the same person who had asked for another page a minute or two before, if it was a logon session that would allow them to stay logged on so that they could sort of introduce themselves to the website in the beginning, and then had this sort of a persistent relationship.

And we've now extended that so that, for example, in many cases you can say, for example with eBay, you can say "leave me logged on for 24 hours." I want to be checking in from time to time. I don't want to have to reauthenticate myself, reintroduce myself, prove who I am with a username and password, every time I click a page. Or even if I'm gone for 10 minutes, I want to be able to come back. So this has become a mature technology, the idea of authenticating to a site and maintaining a relationship with a site over time.

What some clever people recognized was that an advertiser who served ads, a so-called "advertising network" who served ads to many thousands of websites across the Internet, also had cookie privileges, so-called third-party cookie privileges. When you go to a site, and that site you go to, whose URL is up in the title bar, that's a first-party cookie because this is the site you're visiting. But third-party content like advertisements could be served onto the same page. Well, those ads have cookie privileges, so-called third-party cookie privileges. What that means is that even the ad serves your browser a cookie, which is tied to the advertiser's server.

And the problem is that, if you go to a different site entirely, well, the first-party cookie doesn't track because the first-party cookie is tied to the site you visit. The first party is the site you're visiting. But the third-party cookie does track. That is, if you go to a different site, which is served an ad from the same advertising network, since it's coming from the same server, that is, the same advertising network, you'll get the same, you'll have the same cookie transaction. So your browser will send the cookie back to this

other, through this other site, back to the advertising network, and the advertising network can realize that's you. You're the same guy who earlier was over at this site. You're now visiting that site. And that is extended across the Internet so that there's actually now an industry set up to track people and, over time, build a profile of them because the advertisers know what sites you go to.

And so the idea is they infer who you are from the collection of sites you visit, and then they are able to get more revenue from the people they're selling the ads to by saying, hey, we're going to be able to serve ads that are more relevant to the people who are viewing them because we're able to figure out things about them due to the history of where they go on the Internet. Some users object to that. And so that's what's created this whole third-party cookie controversy, where there are cookie crunchers and munchers and disposers and all kinds of technology. Many people turn off third-party cookies, or they flush them routinely in order to prevent this cross-site tracking because they just object to it on philosophical grounds.

Well, it turns out that cookies are only - I would call them, I guess, sort of front-channel tracking, as opposed to side-channel tracking. The biggest problem, though, is that, in the same way we've talked about often, that the Internet, the original Internet technology was never designed for security - remember, it was amazing that it worked at all back in the beginning. And the reason we have so much problem with security today on the Internet, and even with our computers, is that security was an afterthought. In very much the same vein, our use of the web was never designed for privacy. There was a sort of an assumption of anonymity because you never had to declare who you were. You were able to go to websites, and we do today, without ever telling them who we are. People use funny handles to identify themselves instead of their real names. So there's sort of this assumption of anonymity and of privacy.

But the problem is, that's more an illusion than reality. So to say it again, our use of the web, the actual technology of the web, was never designed to enforce privacy. And as often happens in the same way where the Internet in general was not designed to enforce security, and it's ended up not being very, the web, never really being designed to enforce privacy, also isn't very, unfortunately.

Now, when I was doing some background research, I ran across some interesting other instances of side-channel attacks, or side-channel information leakage, that I thought you'd get a kick out of, Leo, as would our listeners. For example, it's possible, it turns out, to identify individual digital cameras from non-uniformity in their optical sensors. That is, there's something called "sensor pattern noise" that individual digital camera elements have, that renders individual ones unique, such that, if you look at a number of pictures from different cameras, it's possible, absent any other information, to determine which cameras took which pictures. Even though they're completely, they're pictures of completely different things, there's just tiny - there's so much resolution now in cameras, so much bit depth, that variations in slight imperfections in the actual optical sensors are enough to identify cameras.

And in a very different sort of approach, because lenses are not absolutely perfect in their production, it turns out that there's a different technology that can be used to identify individual cameras from lens aberration, which can be determined through fancy math, looking at the result of pictures that are being taken with cameras. So there's an instance of information leakage due to something completely different from what the camera, for example, is normally doing. Yet you can, through data processing, you can look at variations among these things which are not the normal information that the device is designed to capture and record, which tells you something about it that the designers never intended.

So a web fingerprint, or a browser fingerprint, is information which is escaping from our use of a browser when we search the Internet, which we're not aware of. So every query which our browser makes to a server contains, by definition, sort of as part of the specification of the way the HTTP protocol works for communicating with servers, contains a bunch of headers. And we've talked about headers in general many times in the past. A cookie is a type of header.

But one of the other headers which is included in every request that a browser makes to a server, is the user-agent. That was something which the very first browser contained. It was sort of a declaration of, like, the browser's name, Mozilla. It might be the browser's version number. Many non-Mozilla browsers, like Internet Explorer, for example, still has the word "Mozilla" in it, last time I checked, because some software just sort of assumed that anything that was going to be surfing the web would have the word "Mozilla" in it because Mozilla was, like, in the very first browser. And so software just sort of looked to see if it was there. And so for compatibility's sake, when Microsoft came along with Internet Explorer, they said, well, we'd better put Mozilla in here, even though we're not Mozilla, just so that we're recognized as a browser.

Leo: A lot of browsers still do that. I always wondered why that was. Now I understand, yeah.

Steve: Yeah. And in fact just this morning I looked at the headers which my browser is adding to every query. And my user-agent header reads: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.2.8) Gecko/20100722 Firefox/3.6.8 (.NET CLR 3.5.30729). The point of that is that that doesn't uniquely identify me and my browser, but it provides a lot of information. Other people will be using different versions of Firefox, will have a different version of the Windows Common Language Runtime. Mine was - RV must be Runtime Version, I guess, 1.9.2.8. So those numbers will vary. Other people using Firefox will be on different platforms, so it won't say Windows. It might say Mac OS X; it might say Mac OS 9; it might say Linux or distributions.

So the point is that, innocuous as it was always intended to be, this user-agent field provides an abundance of information - not unique, not so far, but there's a bunch of stuff there which, you could argue, we don't really need to tell anybody. It's like, whose business is all of that? What purpose does it have? So there's one header.

Another header, also part of every query, is the accept header, which is a way for my browser to say to the server, I'm going to accept the following stuff, the following formats. And so, for example, mine says: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8. And again, there's just a lot of stuff there. It turns out that different browsers have those comma-delimited things sometimes in different orders. There's an implication of the ordering being sort of the order in which the browser would sort of - sort of a hierarchy of priorities. These q's are quality fields. They might differ from one browser to another, or platform to another. So there's still more very specific information which can vary from one user to another. Again, not unique. I'm sure many people have exactly the same accept header. But theirs may be different in a way that the user agent header, the prior one I talked about, isn't.

So now we sort of have orthogonal information. We've got two things that may be independently different, creating a constellation of possibilities that begins to narrow down the field among all the people making queries to the server. And this information is sent out every time we contact the Internet. Every time the browser makes a query, that

stuff goes out. And then there's the accept-language header. That's a separate header. Mine says en-us, so that's U.S. English. But many people, that's clearly language based. So there's many languages spoken all over the world, I mean, and even different subsets of English are being spoken. So that's going to vary from one user to another. So that's a sample of just the simple information, just in the headers, the so-called "query headers" that go out every time we make a query.

So how effective is this kind of information? And we're going to be talking more about much more sort of detailed leakage that occurs. But this is effective enough in aggregate, and what we'll be talking about in a minute, that there are now a handful of companies which have set themselves up to offer robust commercial solutions, sort of as a third-party entity, to companies who have decided they're willing to pay for something beyond cookies. They're willing to pay for what I would call, euphemistically, "nonconsensual user tracking."

There's a company called Arcot that claims it's able to, get this, ascertain the PC clock, the personal computer clock's processor speed, along with other common browser factors, to identify a device. So they've got a means for figuring out how fast the crystal is running in the PC. Again, many people have the same clock frequency on their processors. On the other hand, we know that there's a wide range of processor speeds, down below one gig and up to 1.5 or two or three or higher. So this just provides one more parameter that can be locked onto, that can be used to uniquely identify users, not individually, but it's another parameter.

There's a company, speaking of parameters, called 41st Parameter, that actually looks at more than a hundred parameters from people who are using the Internet. And at the core of their algorithm they have what they call a "time differential parameter" that, get this, measures the time difference between a user's PC's time-of-day clock, down the millisecond, and a universal time reference.

So just, I mean, I know, for example, when I'm sometimes buying things on eBay, I like to swoop in at the last, literally at the last few seconds and put in a bid so that I'm not jockeying back and forth with somebody who also thinks, oh, well, if he's going to bid up, I'm going to outbid him a little bit more. So I'm very conscious of synchronizing my computer's clock to a universal standard. And I notice that from day - I leave my computer on typically 24/7. So after a few days, it'll have drifted a little bit. And so I have to - I resynchronize it to a strong Internet reference in order to have the same clock that's synchronized with eBay so that we're in agreement about when the auction is closing. So again, something like that, just something like how far off your computer's clock is from universal standard gives them one more parameter that they can lock onto.

There's a company, and I love this name, called ThreatMetrix, which claims to be able to detect irregularities in the functioning of the TCP/IP stack, the TCP protocol, the Internet protocol stack, which they say allows them to pierce through proxy servers. Proxy servers is a - we've talked about proxy servers, the idea being that your query goes to a proxy server, which then reissues the query on your behalf. Well, as you could imagine, since your browser's not then directly connecting to a web server, running through a proxy server would tend to anonymize you to some degree. And in fact there have been companies called Anonymizer.com, for example, whose business was to attempt to provide anonymy - anonymy. Anonymity.

Leo: Anonymy. I've got anonymy.

Steve: To provide anonymity that you wouldn't otherwise have, specifically by insulating you from directly connecting to web servers. So, but these guys say we can pierce that.

And then there's one final company, Iovation, which provides device tagging through something called LSO, Local Shared Objects, that we'll talk about in a second - that's a technology that is built into Flash - and what they say is clientless fingerprinting, meaning nothing running over on the client side. And their big claim to fame is they operate a "reputation database" which maintains data on millions of PCs. So they're fingerprinting us and building a repository, a database of millions of PCs. And then they sell this information to third parties. So this stuff is effective enough that commercial companies now exist to sell these as services.

So let's step back and look at this Panopticlick experiment, which was done during the first half of 2010 by the Electronic Frontier Foundation, EFF. The result of this first half year was a paper that they submitted to the Privacy Enhancing Technologies Symposium, PETS, which was held in Berlin just last month, July 21st through the 23rd. During the course of the first half year, this Panopticlick.EFF.org website was visited by 470,161 web browsers, so a little over 470,000 web browsers, just shy of half a million. The code which the Panopticlick site ran in people's browsers and also collected passively from their browser...

Leo: Tell me it's not JavaScript.

Steve: Oh, yeah. Oh, yeah.

Leo: Oh, boy.

Steve: We've got a lot of JavaScript in here. JavaScript is not our friend, unfortunately, when it comes to privacy. So passively, the user-agent field that I talked about before that's just sent with every query, was used. The http, that accept header that I talked about was used. They also looked at whether cookies were enabled or not, which I thought was interesting. Again, not necessarily the cookie itself, but whether the cookie had been disabled. Just a binary. Just yes or no. Because many people, by default, cookies are enabled. So many people who visited had cookies enabled. Many people who visited had them manually disabled.

But what that provided, and we talked about this incidentally, like last week, when we were talking about, answering one of the Q&A questions. Somebody was asking about this notion of entropy. Well, every bit that you add divides the world into half again - those with it set, those with it not. So just noticing whether cookies were enabled gave them one more bit of information.

Then the screen resolution, they ran JavaScript when you visited this Panopticlick.EFF.org site. JavaScript was run as part of what the browser did. Pretty much everywhere you go now, JavaScript is running, when you go to someone's site. So that's not unusual. This JavaScript looked at the screen resolution and reported that back because, again, additional information about the user. Many, many people have the screen resolution of other people. But there's also many different screen resolutions. So it's a nice metric for disambiguating a given user because a given user generally has a fixed screen resolution. That is, if you're using a laptop, you're going to be using that laptop's native screen resolution almost without fail. And most desktop systems aren't changing their resolution

a lot. They've got whatever, especially now that we're in the land of LCDs. Once upon a time, with CRTs, the resolution was a little more dynamic. Now, you generally run at the same resolution as your LCD panel because that's the way you get the right good-looking screen. So lots of people have different resolutions, another sort of data point.

Then JavaScript is also used to report the time zone, which I think is very clever. You know, we've got 24 hours in the day. Time zones are going to be zero to 23. What's your offset from UTC? One more piece of data to collect. And then a huge amount of information was available, again thanks to JavaScript, which enumerated the browser plug-ins and the versions. So all kinds of us have different browser plug-ins, like Flash, like Silverlight, like Adobe Reader, that has a browser plug-in component so that we're able to view PDFs in our browser. And Firefox users probably have a handful of different plug-ins, Firefox add-ons. And each of those has a version number.

And what I thought was really interesting was that these guys recognized the versions were so specific now, and, I mean, people probably are used to hearing me talk about version 1.2.3297.5265, I mean, sounds like a star date. It's so much resolution, so many digits in there, that these guys, the EFF guys call them "micro versions," because they're subversions of the major version. But again, that's information. Many people will have that same version, except not many people will have the entire constellation that I've just run through.

Oh, and the last thing was system fonts. One other thing that differentiates computers is what fonts they have installed. And it turns out that the way that the fonts are enumerated, when you step through them, file system variations, sometimes the fonts come out alphabetically. Oftentimes they come out in the order they were installed. And the installation order sort of tells you a little bit about the history of that machine. So that's going to be different from one machine to another, and generally static. It's not going to change from one time you ask to the next. So aggregating that information, and it's worth noting that there are many other things that can also be locked onto.

I'll sort of wrap this up by talking about other things. The EFF guys recognized and acknowledged that this wasn't an experiment to, like, develop a commercially robust solution. They just kind of wanted to get some idea of, if they did those things, how unique were the visitors who came by? What kind of a fingerprint would that information allow them to build? So browsers without Flash or Java, browsers that didn't have either Flash or Java, 83.6 percent of the browsers that visited their site had an instantaneously unique fingerprint.

Leo: Wow.

Steve: 83.6. No cookies. Not using cookies. Just this other stuff, passively acquired thanks to JavaScript - passively acquired - 83.6. And of those that were not instantaneously unique, 5.3 percent were only confused with a second browser. That is, if it wasn't unique, then 5.3 percent only shared the same fingerprint with one other browser. So what that resulted in was 18.1 bits of entropy, that is, the fingerprint that they were able to obtain essentially gave them 18.1 equivalent bits. And what that meant was that they had - that technology allowed them to disambiguate one browser out of a set of 286,777. That is to say that there were that many bits was equal to 286,777, meaning that a given browser could be pulled out of a set that size without Flash or Java.

Now, most of us have Flash. And as we know, most of us have Java. With the additional help of Flash or Java, which is present in most of our systems, that 83.6 percent jumped

to 94.2 percent. So 94.2 percent of browsers were instantaneously unique. And among those that were not, 4.6 percent were only seen twice. So there was only a confusion of one other browser out of that 94.2. So that brought the level of entropy up to 18.8 bits, or one in 456,419. So hugely discriminatory. Oh, and when you had Flash or Java, then only one percent of browsers had anonymity sets larger than two. That is to say, out of 100 percent of the browsers that visited, only 1 percent were not unique. Only 1 percent would have been confused with more than one other browser.

So this is phenomenal. I mean, you don't need to say anything to an advertiser or anyone with an interest in tracking you, which never - no one is assuming it's 100 percent. But here we're at, like, 99. If you add 94.2 to 4.8, what do you get? You get 99 percent actually, yes.

Leo: Good enough for most, I would say.

Steve: Yeah. So, and that follows because that 1 percent was not specific to less than two browsers. So what they learned was that, without using cookies, with no cookies at all, just looking at passive browser headers and with the help of JavaScript that was able to enlist the help of Flash and Java - and Flash and Java, by the way, were used for the system font enumeration. JavaScript was able to be used for returning screen resolution, time zone, and enumerating the browser plug-ins and versions. So without Flash or Java, that got them to the 83.6 level. Flash and Java, which added the system font enumeration, brought them all the way up to, if you're willing to go for an instantaneously unique browser, that brought them to the 94.2 percent.

Now, what they did recognize was that fingerprints are going to evolve over time. That is, my system, when I went to Panopticlick middle of this period, probably back in March, would have had a given fingerprint. I was one of those many browsers that went. But then I updated to a new version of Firefox. Well, that would have changed my fingerprint somewhat. Or NoScript came out with a new version, so I updated that. And that would have changed my NoScript plug-in. But what they recognized was, because they weren't just mashing all this together, that is, they didn't take all that and, for example, hash it into an opaque token. They kept all that separate, which allowed them to track the changes, that is, they knew when I updated my version of Firefox because only that one thing changed.

Leo: So they could update the database.

Steve: Yes. And what they found was that, now, here they did use third-party cookies to create a persistence among visitors because they were just doing this for collecting data. So there were 8,833 browsers which accepted cookies and which returned several times during this testing phase over a period of more than 24 hours. Of that 8,833 browsers where they were able to give them a persistent cookie, and that's what allowed them to recognize uniquely, guaranteed uniquely, when that one browser came back to their site sometime later, more than a day later, over the course of several times, 37.4 percent of the time there was a fingerprint change. And get this: They were able to guess correctly, not taking advantage of the cookie, but just looking at the evolution of the fingerprint, they were able to lock on and hold onto the person 99.1 percent of the time. They guessed correctly about what change the fingerprint had made, and they were able to still lock onto the return visitor only using their fingerprint. And their false positive rate of guessing incorrect was 0.86 percent.

So what that told them, they used the cookie in order to accurately track people. So then they were able to look at the fingerprints and track the changes. And what they saw was that little tiny changes were being made over time, and that they were able to move, to sort of move forward with versioning of these things in order not to have sort of a synchronization lock lost, just using these features that they were tracking.

So, let's see. Explicit channels, we know about explicit channels as opposed to side channels. The HTTP browser cookie, standard web browser cookie, is an explicit channel for tracking. Now, less well known, but arguably still explicit, is the Adobe Flash so-called "supercookies." Those are regarded as supercookies because we now know many sites are using them. There are some commercial services, in fact, which are now falling prey to lawsuits because people are arguing that they've deliberately flushed their cookies because they don't want to be tracked, yet there are businesses that are selling the reconstitution of deleted cookies by using Flash cookies.

It is possible to disable Flash cookies, but they are enabled by default, and there's no convenient user interface. The browser interface doesn't allow you to block Flash cookies. You've got to go to Adobe and go through some hoops in order to find the UI, which is not easy to find, in order to get there and turn this off. Which means it's a high bar that most people don't climb.

So side channels, aside from those two explicit channels, we've got the standard leakage of browser queries. And remember, none of this was designed for privacy. It was just designed to work. And consequently, it doesn't really give us much privacy. So we know about the accept header, which the browser sends out, and the user-agent. Then there's this micro version information where we're, like, providing too much, you could argue, versioning information because it makes individual systems very unique. And this information is available to servers, and whether cookies are enabled or not, not even what the cookie's value is. But as we said before, whether cookies are on or off divides the universe into those with it on and those with it off. So if all other things, if all other fingerprinting information was the same, one person might have turned their cookies off, and that would differentiate them from the person who hadn't.

Leo: Geez. You can't win.

Steve: You know, exactly. Whether images are enabled or not. Some people surf with images off. Some people fake their user-agent because they think they're being clever. Except it turns out...

Leo: That's worse because you have an uncommon user-agent.

Steve: Exactly. It turns out that there's other information about things like - get this, Leo. Different browsers issue the query headers in different sequences. So even the sequence of the browser headers tells you - it's a way of fingerprinting the browser. So if the sequence of the headers doesn't match the user-agent, well, that tells you the guy's got a spoofed user-agent. So, bing, there's another piece of data about this guy.

Leo: Really, if you think about this, it's almost obvious. And really the only reason this comes up now is because there's such demand to track people.

Steve: Yes.

Leo: I mean, of course your computer's unique when it goes out in the world.

Steve: Yes. There are just too many things about it that are not exactly like somebody else's computer - how many screens you've got, what their resolution is. Even, and we've seen this before, there's a privacy problem with CSS and with browsers because they color the links differently, whether you've visited things or not. It turns out that scripts are able to determine the link coloration, which is one other piece of information. It's even possible, by looking at what your browser fetches, to infer what's in its cache. And what's in your browser cache...

Leo: Oh, wow. Oh, my goodness.

Steve: Uh-huh. What's in your browser cache is different from what's in somebody else's.

Leo: Oh, you used that picture? Ah, well, we know it's not you, then.

Steve: Exactly.

Leo: Wow.

Steve: If your browser makes the request, it's because it doesn't have it. And once it does, it doesn't ask for it, if you give it a link. And we haven't even talked about IP. We know that IPs are not unique. But they're certainly less than random. Many people notice that their IP drifts around. Maybe, you know how IPs are four bytes, well, the first couple bytes, if you're using a given Internet service provider, they never change because that ISP is assigned a big block of IPs, but it's only the least significant bytes that change.

So there again is another valuable piece of information, doesn't uniquely identify you; but when combined with everything else, it provides many more bits of information. And the EFF didn't even use that. So that's not even part of what they, I mean, that would have been a bonanza of additional disambiguation, had they taken advantage of IP. And then things like clock skew. They didn't use that, but we know that there are people who do. So one of the interesting things is that certainly we know there are people who do not want to be tracked. Paradoxically, as we just noted with the example of faking your user-agent, which some people do, if you do things like that to try to obscure yourself, you're actually identifying yourself. You're pulling yourself, you're creating something which is different from everybody else who is otherwise just like you, which now identifies you, even though you did something trying not to be identified.

Leo: So it sounds like the real problem is how much is available through JavaScript calls, how much information about the machine. It seems like a lot of that doesn't need to be revealed.

Steve: Correct. I would argue, in the same vein of NoScript being valuable because it prevents scripting unless you know you need it, the fact is not running with JavaScript certainly enhances your privacy also because, to the degree that there are companies that are feeding JavaScript through ads, and ads run JavaScript just like anything else, to the degree that there are third parties that are injecting JavaScript into your browser session for the purpose of collecting this information and using it for non-cookie-based side-channel privacy tracking, not having scripting is a benefit for privacy.

Now, what's interesting is that people who, like, say, well, I want to increase my privacy, so I'm going to flush my cookies, the problem is, as we now understand, cookies are, if they're available, very powerful. I mean, they were built for tracking. So many people turn them off, or many people flush them from time to time, thinking, okay, now in flushing my cookies, or in deleting them, I'm starting with a clean slate. Not so.

The problem is that, if you had cookies enabled, and you were on the 'Net, then someone somewhere is building a sophisticated fingerprint AND the cookie, that is, they're happy that you're accepting cookies, but they're not only relying on that. They're also building one of these fingerprints on the off chance that you're going to delete the cookie.

So imagine the sequence. You're cruising around. And of course Double-Click, for example, is serving ads and maybe running some scripts to do what they can to track you. You then decide I'm going to delete my Double-Click cookie because I want them to forget who I am, to lose track of me. So you delete your cookie. You shut down your browser, and you restart it, and you go back on the 'Net. The instant you hit a site that is served by Double-Click, Double-Click sees you don't have their cookie anymore. But they've got your fingerprint, which hasn't changed over the course of that shut down your browser and restart. So they've still got a hold of you and give you a new cookie, tied to you just as much as the prior cookie was. So the lesson here is, if you are - well, okay. Part of the lesson is just give up.

Leo: Well, yeah, they're saying in the chatroom right now, well, we can't even watch this show without using JavaScript.

Steve: Right. So in my conclusions, in my own notes here I said, for now, maybe don't worry about it. These things are probably going to get better. I would say the takeaway from this podcast is appreciate what's being done. Understand what's being done, and behave accordingly. You want to recognize that this is what's going on, unfortunately. Also unfortunately, our computers are just bleeding information about us as we use the Internet. I mean, it's pouring out of every contact we have with websites. All of this is available. So rather than imagining that you are not trackable, or that you're achieving something from deleting your cookies, recognize that you've lost that battle.

Leo: It's too late.

Steve: It's too late. And just - so set that expectation as being the case and behave accordingly. Don't do things where it's important for you to have anonymity that you actually don't have. But I would also suggest that resistance is not futile, that putting up some barriers is a useful thing to do. If you really were serious about not being tracked, what we have learned from all this is don't just change one variable. Don't just delete a cookie, because everything else is still there. Don't just change your user-agent. Don't just change your screen resolution. Don't just change one thing because the technology

that has been developed will track small changes. It'll jump just as the EFF demonstrated. They're able to straddle small changes. What you really need to do is make a big change to your system at once, like download a bunch of updates and apply them all at once. And suddenly your system looks like a very different machine than it did before. Maybe change your screen resolution at the same time for a while.

Leo: Just randomize everything.

Steve: Exactly. Try to make the largest change you can, and that'll throw off anybody who's trying to track an incremental fingerprint over time. But this is truly happening, this kind of side-channel privacy leakage, I mean, where they're looking at how far your time-of-day clock is off and actually collecting that information and using that as one more piece of data to differentiate you, to disambiguate...

Leo: It's pretty amazing.

Steve: ...you from all the other people on the 'Net. Isn't that incredible?

Leo: It's a neat exercise. I like that part.

Steve: It's a hack, it's great hack.

Leo: Yeah. I find that fascinating. Do we know if people are using these techniques in...

Steve: Yes.

Leo: They are.

Steve: Commercial companies have stated this is what they're doing. We know that this is being done.

Leo: Wow. Well, I mean, I guess people who say "privacy is dead, get over it," have another bullet in their gun.

Steve: Yeah.

Leo: And you can't not use Flash or JavaScript. Eventually you're going to have to turn it on at some sites.

Steve: Oh, I, for one, I mean, I'm annoyed I don't have it on my iPad.

Leo: Right.

Steve: So here I'm annoyed that I don't have something trying to tattle on me.

Leo: Yeah, right. Oh, Steve. Always fun to hear these stories. I don't know what we do about it. Really, the key was when they designed JavaScript they did a very, very poor job because they allowed it to query too many system variables.

Steve: Well, it's code. And it wants to be powerful enough to do, like, things that Google is doing, these amazing things with JavaScript. But that also means that you can do things like report your screen resolution.

Leo: Right.

Steve: And there's some personally - it's not - it doesn't - now, none of this identifies you personally. It doesn't know your name, your street address or anything else. But we've already...

Leo: Doesn't need to.

Steve: Exactly. It's trying to profile you to determine what your profile is. And we do unfortunately know that there are other means for them to figure out who you are because this fingerprint that you carry around, you have when you're on eBay, when you're on your banking site, when you're other places. Now, banks are apparently using this to help with fraud prevention. They're using these fingerprints on our behalf as additional verification that we are the same person that we said we were last month. So you could see a positive benefit to it, unless there's a relationship where the bank or some other organization that you have identified yourself in the physical world to, they could be selling that information back to these aggregating tracking companies. And we know that's been done in the past also. So it may not be that we're as anonymous as we wish we were.

Leo: There you go. Steve Gibson is the guy in charge at GRC.com, where you'll find no tracking cookies of any kind, I guess. Who knows?

Steve: Well, only for the purpose of illuminating what's going on for our users. I do have, and not yet public, a very nice third-party cookie monitoring facility that allows you - that shows you what your browser settings are. And that'll be going public soon. The technology's been finished for quite a while.

Leo: Oh, that's cool. GRC.com. Lots of free stuff there. Of course there's also SpinRite, Steve's bread and butter, his day job, the world's best hard drive maintenance and recovery utility. You can get it at GRC.com. You can also

participate in the show in a lot of different ways. We've got 16KB versions there of the show for people who are bandwidth-impaired; transcripts. You can leave feedback or ask questions at GRC.com/feedback. In fact, next week Steve will be answering questions. So that'd be a good time to do that. And I guess that's...

Steve: And a bunch of freeware and good stuff and...

Leo: Freeware, really good stuff, yeah. GRC.com. Follow Steve on Twitter. He's there, he's tweeting, and his handle on Twitter is @SGgrc. He also has an account for pad and tablet-related stories, @SGpad. And then the corporate account is GibsonResearch. When he goes, he goes whole hog. Steve, thanks for joining us. We'll see you next time on Security Now!.

Steve: Thanks, Leo.

Copyright (c) 2006 by Steve Gibson and Leo Laporte. SOME RIGHTS RESERVED

This work is licensed for the good of the Internet Community under the Creative Commons License v2.5. See the following Web page for details: http://creativecommons.org/licenses/by-nc-sa/2.5/