SECURITY NOW!

Transcript of Episode #119

PayPal and DoubleClick

Description: Steve and Leo dissect the "Links" on PayPal's site with an eye toward reverse engineering the reason for many of them routing PayPal's users through servers owned by DoubleClick. They carefully explain the nature of the significant privacy concerns raised by this practice.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-119.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-119-lq.mp3

INTRO: Netcasts you love, from people you trust. This is TWiT.

Leo Laporte: Bandwidth for Security Now! is provided by AOL Radio at AOL.com/podcasting.

This is Security Now! with Steve Gibson, Episode 119 for Thursday, November 22, 2007: Third-Party Cookies.

This is Security Now!, ladies and gentlemen, the show where we answer your questions.

Steve Gibson: But not this week.

Leo: Not this week, no, no. You didn't know what the question was. Help, help, I'm in security denial, I'm in security limbo. Steve Gibson is here from GRC.com, the security wizard. And we've been having fun talking about all sorts of different topics. Are we going to talk about DoubleClick and PayPal today? Is this the day?

Steve: Yes, yes, yes. And I'd have to say, Leo, this is probably the most requested topic. We've never really had a most-requested topic. But when I was scanning through all the email for last week's Q&A #28 for last week's show #118, many people were saying, hey, you never followed through on that. You didn't, you know, whatever happened with that? You said you were going to talk about it, blah blah blah. Well, of course what happened was the Perfect Paper Passwords exploded into something much bigger than I expected, and so we gave it a number of weeks. And that pushed our coverage of this interesting PayPal/DoubleClick relationship into the future, to today.

Leo: Well, I know that you stirred up a hornet's nest when you mentioned this, so we'll find out what it's all about. Before we get to that, do you have any errenda or - errenda - addenda or errata?

Steve: I don't, but I got one really nice email regarding SpinRite that I wanted to share with our listeners because it's from a guy who, as you'll hear in listening to this mail, has a good grip on PCs and a sense that he knows how to fix things. Anyway, he wrote on November 19. I got his note forwarded to me through my tech support guy. His name is Frank Barker, and he says, "I had a situation where a Windows XP PC would no longer boot. This PC contained more than 50 gigabytes of pictures, videos, and other very important data which, of course, was not backed up. Not a good situation. I tried various commonly used techniques to correct the booting problem but was unsuccessful. Normally I would simply add the problem drive as the second drive in another good PC and copy the data from it. In this case, Windows on the second PC would not even recognize a valid partition on the problem drive. Each time I attempted to access it, Windows asked if I would like to format the drive. Obviously an unwelcome question. After nine-plus hours of wracking my brain, attempting various methods and a few utilities, I was able to recover only a few meg of data. Basically nothing."

Leo: Oh, boy.

Steve: So he says, "Mentally exhausted and borderline desperate, I decided to spend the money and download SpinRite." Clearly his last resort is how he was thinking of it.

Leo: Well, and I hope he hadn't screwed up the drive with all the other recovery attempts.

Steve: Yeah, it sounds like probably it was just him trying to read from it, you know. But you're right, it's a little risky when you start doing other utility things. So he says, "I decided to spend the money and download SpinRite. Although I was skeptical of any success, I kicked off SpinRite at Level 2. SpinRite ran for nearly two hours and found a few defects on the hard drive, some recoverable, some not. Nothing SpinRite displayed particularly led me to believe I had gained much, if anything. Nonetheless, I again added the problem drive as a second drive in an existing system. I again attempted to open the drive through Windows Explorer, except this time I was shocked and amazed to see the folders and files in the root folder of the drive. Still skeptical, I drilled down into the folder structure to where the data was. To my surprise, the navigation continued, step by step, to work. And I was able to copy all of the data from the problem drive. I can't begin to describe my relief. After so many hours of struggling, loss of nearly all hope, and then full access to the drive and all the files it contained, I was stunned. Thanks for developing and providing a product that delivers on the claims."

Leo: That's great. Wow.

Steve: So I was really, I mean, this guy was about as hard to convince as anyone could ever be. He tried everything else he could think of. And he said, oh, okay, fine, [indiscernible].

Leo: I give up.

Steve: I don't think it's going to work, but I'm going to try it anyway.

Leo: So it sounds like probably there was a flaky sector in his file allocation table maybe, or his partition table?

Steve: Yeah, it would have had to have been a serious problem because Windows was unable looking at the drive to see that...

Leo: Didn't even think it was formatted.

Steve: ...there was even a - exactly, that there was a valid file system. And so it said, hey, you know, here's an empty drive, do you want to format it? And he said, uh, no, it's not what I had in mind, actually.

Leo: But, you know, all it takes is a bad sector in the partition table that Windows can't read, and sure it would have that reaction.

Steve: Yeah, the NTFS file system is supposed to have redundancy and other ways of getting around things. What bugs me, and I understand this because obviously he didn't believe that this utility, that SpinRite was going to provide any value. So he's also someone who certainly wasn't running SpinRite on this machine prior to it getting into this shape. And one of the things that we've seen through all these testimonials that I've shared with our listeners over the last couple years, is people are always waiting until it's almost too late, and then SpinRite brings them back from the precipice.

Leo: Same thing with backup. People don't do any of this stuff until it's often too late. So how often would you run - I mean, I confess, Steve, I have SpinRite, I don't run it all the time. How often should I be running SpinRite?

Steve: Well, see, then again, that's human nature. The good news is there's enough plasticity in drives that SpinRite is able normally to bring something back from the precipice. And I think that's pretty much what we all rely on now is, oh, well, if something's going to happen, SpinRite will fix me. It's like, okay, well, let's hope so.

Leo: I back up like crazy, too. I mean, that's the main thing.

Steve: Actually, you really do. And from my own standpoint I'm Mr. RAID. I mean, nothing anywhere is critical that I don't have, essentially, redundancy thanks to RAID. So...

Leo: I have RAID 5 backup. So I don't have RAID 5 online, but it's offline. The backups all go to RAID 5 drives.

Steve: Right.

Leo: But our web - actually I take it back because our web servers, I mean, I have two dedicated servers. Those are both running RAID 5 with Raptors for speed and redundancy.

And I back up. I still back up like crazy.

All right. So we're going to get to this issue, this fascinating issue of why PayPal gets its files from DoubleClick. And I can't wait to hear what you've found out. All right. So this all started - was it an email that you got, or how did you find out about this?

Steve: Well, okay. It was really interesting. This was one of our questions from a while ago, at least a month or two ago. One of our listeners was using PayPal, and I think he was going to the virtual debit card is the link he was using. And he couldn't get there. The PayPal website to him seemed to be broken because he was clicking on the link and nothing happened. And through some analysis of his own, he determined that his hosts file which he had in his machine set up to block access to a number of sites he did not want his machine to visit under any circumstances, the hosts file had an entry for DoubleClick.net. And that entry was preventing PayPal, the PayPal website from functioning.

And so he wrote to us and said hey, you know, what's this about? And it immediately raised a red flag for me. I don't think I had noticed before, looking carefully at the URLs as I hovered my mouse around the various links on PayPal, what the URLs were aimed at. But what was immediately apparent is that there is a relationship between PayPal and DoubleClick that goes beyond the typical advertising model relationship. And so we're going to pull back and look at the way this normal advertising relationship works, sort of do a little quick refresher in the way browser URLs and headers work relative to cookies and tracking and so forth, and then expressly look at what it means that PayPal's links do not go to PayPal, many of them actually go to DoubleClick. And so it demonstrates, while we're not able to say definitively this is the data that is being shared, by taking a look at what's going on there's a lot that can be inferred. And there is only really one good reason that this is being done.

So stepping back from this a bit, the way - all Internet users are certainly familiar with this wacky http://www.domainname.com and the way that operates. There's a lot that goes on behind the scenes that we've talked about in prior episodes of Security Now!. I'm going to sort of pull specific aspects of those that are relevant to this particular issue and sort of refresh our listeners' awareness of what's happening.

First of all, when you send a URL to your browser, you give your browser a URL, it looks up the IP address of the domain in the domain portion of the URL and establishes a connection to the remote server at that location. So www.paypal.com, for example, that is translated by DNS into an IP. And probably in the case of PayPal, like the large websites, they'll actually have a large pool of IPs which the DNS server rotates among so that the load is automatically spread among all of the machines that are set up to receive incoming traffic. So www.paypal.com, probably every time you look it up you may get a different IP. That's often the case. Or it might be a single IP and then there's some sort of a load-balancing system on PayPal's end that is then distributing incoming queries all to a single IP to, again, a large - a farm of individual machines. But so one way or another, your browser gets a connection to a machine at, for example in this case, PayPal.

Now, several things happen. If you had visited PayPal before, that is, if your browser had visited PayPal itself and received a page in the past from PayPal, there are headers that are not part of the page. That is, it's sort of like it's preamble information that has sort of metadata about the page, things typically like how long this page should be considered valid before it should be considered expired by the browser. That allows your browser to keep the page. For example, if you went to a different page and then you hit your back button, the browser would see that it has that page recently received. And if it's recent enough, if it's allowed to be presented to you as non-expired, then the browser is able to save time by simply presenting that page from its local cache rather than having to go and fetch it and all of its contents again. So there are some nice optimization capabilities built into the whole web system.

So there's a bunch of these, this metadata. For example, there is a tag that indicates what flavor and version of HTML you're using. In case there is evolution in HTML, it would help the browser to interpret the rest of the data. So this is stuff that you don't see, but it's at the beginning of the response for any sort of a web asset, whether it's a page, an image, other types of media and so forth. There's always these browser headers.

Well, one of the headers in there was designed originally by the Netscape folks to allow a stateful relationship with the server, meaning that because each page query comes in with a connection to a server and sort of stands alone, it's difficult to associate a person on a website who's moving from page to page with themselves. That is, oh, look, this person is somebody who asked for that page, now they've asked for this page. Because, you know, there's a world of people all clicking on links, and they all look like they're separate. There's really no way to maintain a relationship with someone as they move around a website.

So Netscape invented this notion of a cookie, the idea being that it's just some sort of a blob, it's considered an opaque token, meaning that it doesn't necessarily have any real meaning except to the server that issues it. So one of the items in this metadata that comes in, never seen by users when you're normally looking at a web page, one of them is this thing called a cookie, where the server offers it back to the browser, and the browser in normal configuration will save that cookie as, essentially, sort of as a local ID for itself. And then any subsequent queries that are made by the browser to the same domain, that is, even if it goes to a different IP or to a different machine, the browser only concerns itself with, for example in this case, www.paypal.com.

And so the cookie is stored locally in the user's machine, in their browser memory, tagged with www.paypal.com. So that if, again, any query is made to that domain name, www.paypal.com, the browser is always looking through this library of cookies to see whether it's got one that it had previously received from a prior query to PayPal, if it has one. If so, and if it's not otherwise configured not to do so, that is, in all browsers' default configuration they offer that cookie back. In other words, we see that we're asking for http://www.paypal.com.

But queries, that is, queries out to servers also have metadata as part of what's called the "query header." And there are things in there, for example, in the old days browsers used to say, oh, my screen is 1024x768. The idea was that, oh, that might help the server to give you back a page that would fit your screen better. So there are things that your browser is also sharing that we don't see going out. Again, it's part of query metadata on the way out. We see the URL. There's actually more stuff that is being sent by the browser. And one of those things is this cookie. If it matches the domain name, and if the browser is in its default configuration, it'll say, oh, look, once upon a time you gave me, you server, PayPal.com, gave me this token. And I don't know what it is or what it means, and I don't have to. But the deal is I'll give it back to you. And so what this allows is it essentially - this is one of the core enablers for a much richer experience on the World Wide Web. We are able to, quote, "log-on" to a remote service, whether it's PayPal or Yahoo! or MSN or Gmail or whatever.

And once we authenticate ourselves, we receive a credential, in the form of a cookie typically. And the browser keeps offering it back every time it asks for other things from that same site. And that allows our log-in to be stateful, that is, it allows us to move from page to page and be remembered as we're sitting at this computer, we logged in X number of minutes ago. And so as long as I'm active here, keep me logged in. If 30 minutes transpires and nothing has happened, then the remote server is able to say, okay, we're going to expire his log-in. And so if you then come back to your computer and try to use the site again, even if you left it up on your browser, many people have had the experience of the thing saying, sorry, your log-in has expired, please reauthenticate.

So all these mechanisms are enabled by this notion of some sort of state being saved. So this was all sort of cool and good. Then unfortunately some clever people back in the - I was going to say the dim past of the Internet, but it's only been a decade - someone figured out that there was another way cookies could be used, which was never intended by the Netscape folks, but neither was it explicitly prevented. And so browsers did this. The idea was that, as advertising began to surface, sites would not be presenting their own ads. They would go to a third-party service, and that third-party service would populate the page with ads. So, for example - GRC's a bad example since I don't have any. But, you know, many sites get some revenue because they're willing to host advertisements on their pages.

Leo: Well, we've got a banner ad on TWiT.

Steve: Yeah, exactly. And presumably...

Leo: And that usually doesn't come from, in this case does not come from our server, it comes from Podtrac's advertising server.

Steve: Okay, exactly. So, for example, in that case, when someone brings up the TWiT.tv page, the HTML for the page comes to their browser. And part of the HTML contains a URL to Podtrac, meaning it says in this area of the page go to the Podtrac advertising server to get the image to put here.

Leo: It's actually - it's interesting. And in our case it's a little bit of JavaScript. And there's a good reason for that because the number of impressions are counted, and that's how we get paid.

Steve: Well, and that was the cleverness about this model is that the advertisers, and of course DoubleClick.net, DoubleClick Corporation, was an early one in this game. The idea was that a popular site would have many people looking at its pages. And so every time one of those pages was displayed, the user's browser would go get content for the ad, not from the site they were visiting. That is, not from PayPal, not from TWiT, but from a different location that was referenced by the page that was being displayed. And the beauty is, the idea being that the more popular the site, the more page impressions people were seeing, the more ads would get pulled from this third-party server. And third-party is the key because what we've been talking about here so far with a user and their browser is considered a first-party relationship, meaning this is the site I've gone to. I've gone to PayPal.com. That's where I am But PayPal, the page that comes up has a reference to a third-party that is not the one, you know, is not PayPal, it's some other server. And so my browser, the user's local browser, goes and fetches the content under direction from the first-party page. It goes and fetches the content from this third party.

Well, the hook here is that that fetch, that query and response is also cookie-enabled by default. Meaning not only can PayPal track us as we move around PayPal's site, but DoubleClick, someone with whom we have no relationship, we have not gone to DoubleClick's server, we may not particularly have any interest in anything DoubleClick has to offer us. But our browser has now a DoubleClick cookie which it is carrying. If our first contact with DoubleClick did not offer a cookie, DoubleClick's server sends one back. Which, again, every browser currently set up defaults to allowing this behavior. It will carry DoubleClick's cookie.

So whereas PayPal uses the first-party cookie to track us as we move around PayPal's site, if many different sites on the Internet all have contracts with DoubleClick, then their pages are all referencing DoubleClick. And because of the so-called third-party cookies, that is, cookies that are offered by not a site we're visiting, but by a site whose content is coming to the pages of the sites we're visiting, those third parties are able to track us across the entire Internet. And so it's not just a single website because a company like DoubleClick may have and does have contracts with a huge number of companies and websites worldwide across the entire 'Net. Our browser will continue this association with a company like DoubleClick.net through its cookie and through all the ads that we receive from these sites. Now...

Leo: I have to say, I mean, this is how it works, and this is normal. Ours, actually it was Podtrac, they go through AdvertPRO, which is a business that does this, just like DoubleClick, I guess. And that's the way it has to happen; right?

Steve: Well, okay. So a little more about what goes on. There's another one of these metadata items which is called the "referrer field." Remember, we talked about like there's an expires field, or an expires header, that tells the web browser how long it's able to keep the page, what the conditions of caching the page locally are. And there of course are the cookie fields. One of the things that goes out with a query, that is, from the browser to a remote server, is when a page comes in it contains, as we were saying, pointers, URLs, to other assets - images, pictures, media, other chunks of text, various things, which it then makes secondary requests in order to get all of the images to make the page complete. Those requests are made to the URLs specified by that page.

But part of that request contains a referrer field, meaning it contains the URL of the page which contains the pointer to this asset. Meaning that it sort of knits all this together so that somebody receiving the queries that result from the display of a page also obtain as part of that query the URL of the page that made that request. And that's where this tracking comes in because that means that, for example, from the vantage point of DoubleClick, they are receiving queries for ads from all over the 'Net. They are also receiving cookies with those queries in their default condition, their default configuration. And they are receiving referrer data, meaning they also know where all of these users are going.

Now, so far this is a completely anonymous relationship. That is to say that there has at this point been no explicit identification leakage. So DoubleClick knows that an anonymous person, an anonymous browser received a cookie on a certain date for the first time. And no doubt DoubleClick's got mega databases. Hard disk drives are not expensive anymore. So DoubleClick knows that an anonymous machine received its first DoubleClick cookie on a certain date. DoubleClick then knows everywhere that machine has gone subsequently on the Internet for sites which host DoubleClick ads because every time that machine brings up a page that contains a DoubleClick ad, again with all things in their default settings, that page makes a query from DoubleClick returning the same cookie that it received. And DoubleClick is able to evolve these cookies over time and track their changes.

So it's basically this third party has locked onto this anonymous browser and is able to compile and does compile, we know that this is what these companies are doing because they brag about it. They say that they're able to profile users anonymously and build up some idea of who they are, ostensibly for the purpose of presenting them with more relevant ads. I mean, when I first heard, it's like, okay, well, I'm not sure I like the idea of anything tracking me. But if the ads are more about things that would interest me, they would be less annoying, presumably, on pages. And I've seen this concept work. TiVo, for example, has this notion of automatically recording shows that are similar to the things you've explicitly said or you've given "Thumbs Up" to in the past. I often stumble on something when I'm at Amazon because Amazon has this notion of, oh, look, people who purchased the book you just purchased also purchased these books. Maybe this would be of interest to you, too. So there are ways of networking and aggregating and making this sort of relationship work.

Leo: Yeah, but there's a big difference there. You have a relationship with Amazon and TiVo, and they're the ones doing that.

Steve: Exactly. Well, and the other thing is, there are ways that we know information, that is, anonymous information can leak into a third party. The classic way is when we used to fill out forms. And this has largely been deprecated now, so that this is not happening to the degree it used to. But it used to be, when you would fill out a form you'd put your name, address - back in the good old days before everything became really problematic on the Internet - your name and address, your telephone number, your email address, whatever. And you would press "Submit" because you were submitting it to a site you were visiting.

Well, many people, certainly the veterans among us, will remember that information used to be in the URL. It would be - you would see http://www.paypal.com or whatever site dot com, and then a page or two, and then a question mark. The question mark was this delimiter that said, okay, stop the URL now. Everything after the question mark are parameters. That is, this was the means for data going in the other direction. Remember that when HTTP was originally designed and the web was conceived, we were going to be clients, they were going to be servers, and it was a one-way relationship. We would click on links, and we'd see these static pages. There was no notion of us sending information back to servers. That came later. Of course it's critical for everything we do now on the World Wide Web. But the notion was this whole web was going to be read-only. We would just be getting these pages and happily browsing around this huge Internet.

So the question, the dilemma, was how do we get something back? How do we send something back to a server? Well, what we were always sending to the server was the URL of the next page we want to bring up. So the smart HTTP folks said, hey, we can add data at the end of the URL. We'll put a question mark there to say, okay, this is the end of the actual page specification. Anything afterwards is parameters, it's data. And so it was a clever way of sending data back to a server. The huge problem was that that URL is the referrer, meaning that, if I've submitted a form at a site where the page that came up next, that is, I fill in the form, I push "Submit." The page that comes up next has an ad from, for example, DoubleClick.net. Well, my web browser, with the best of intentions, knows it needs to go get the contents of this image from something called DoubleClick.net. And, oh, what do you know, it's already got a cookie for DoubleClick.net. So it sends the cookie. And as part of the HTTP specification, it fills in the referrer field with the URL of the page.

The problem is, the URL of the page contains the data that I submitted in the form - my name, my address, my phone number, my email address, whatever it is I submitted was tacked onto the end of the URL. And that's part of the referrer field. And it was well known years ago that these third-party aggregators were anxious to know as much about us as they could. And there are confirmed instances where people would go to web pages, fill out, I mean, passively - sorry - passively look at web pages, fill out no information at all, and then receive a telephone call from a telemarketing firm that knew who they were, had their phone number, and knew what page they were browsing.

Leo: This happened to a friend of yours.

Steve: It actually did. It happened to a Canadian reporter who was freaked out by this. He went to the New York Philharmonic website just to see what the calendar, what the schedule of upcoming concerts was because he was going to be coming down from Toronto to New York for a trip. And he browsed around, saw what was going on, turned off the computer, went out into the back yard and was doing some gardening. The phone rang, and this was the Philharmonic marketing company that said, hey, we understand that you're interested in the symphony. We wanted to make sure you knew of a special offer. Well, he was stunned that pushing no buttons, filling out no forms, somebody knew something about him that was clearly, he felt, a breach of his privacy.

So, okay. So that's the whole scenario of how tracking works among websites, how data can leak back. Now, that was such a problem that a different way of sending data was created. And that is, there's a way of sending data that is not in the URL, using what's called a "post" as opposed to a "get" request. So it's a different way of sending the data that does not pollute the URL with the contents being sent back. And that does blind third parties from being able to access data through this URL leakage. Of course the other thing that's happened is that savvy people who are made uncomfortable by this whole notion of third-party cookies have started disabling third-party cookie tracking. All web browsers - I know that Safari does, I know that IE does...

Leo: Firefox doesn't.

Steve: It's weird. You have to go through some real hoops in order to disable third-party cookies in the most recent version of Firefox. 1.5 had it in the UI.

Leo: The rationale the Firefox folks gave was that it never worked, and so they didn't want to expose it in the preferences because it doesn't work. Doesn't do anything. I don't know if it doesn't work because Firefox can't do it or because - I think my impression was they believe that it's impossible to block third-party cookies through the browser for some reason.

Steve: That's not true. What they were saying was it's an imperfect solution. I don't understand why they did this. It's the dumbest thing I can imagine. I mean, essentially they had to have been saying it is impossible to prevent all kinds of third-party leakage; therefore we're not going to do any. Which is completely contrary to everything we know about security. We know, all of us know who've been listening to this podcast, that security is not black and white, that you use multiple layers, you do as many of the best things as you are able to, recognizing that none of them is perfect. But, you know, raising your defenses as high as you can is a good thing to do. So, yeah, I mean, so there is a way to block third-party cookies.

So we ought to - basically the idea is that all browsers that are currently configured will offer back a cookie to a site other than the one you are logged onto, that is, that your browser's query has gone to. So, for example, in this picture I've been painting with PayPal, by default, if PayPal were to display a DoubleClick image, then my browser would give back a DoubleClick cookie to DoubleClick if it had one which it had picked up from any prior contact with DoubleClick.net in the past. It is possible to go in and disable that explicitly. You're able to say, in IE and in Safari and in Opera, and there is a way under the new Firefox, although you have to edit some funky data, there is a way...

Leo: Use the about:config setting in the Firefox.

Steve: That's exactly right. And so what that does is that tells your browser that you're somebody who is privacy conscious. You do not approve of the idea that a site you're not visiting could have a cookie transaction with your browser. Okay. So...

Leo: I still am a little angry at Firefox. If they say it doesn't work or it's an imperfect solution, but they still don't offer it, that's odd.

Steve: I completely agree. And I think there are probably other means of circumventing this. For example, you could probably use scripting in order to maintain some sort of stateful relationship. So the problem is, we've got a very powerful and ever more powerful protocol with the whole web browser experience. And the notion is, if somebody wants to do this, they're going to find a way. Well, that brings us to the whole point of this podcast, which is that PayPal has found a way to explicitly force a cookie relationship between us and DoubleClick, and there is - except for breaking PayPal or being very diligent about cookie management, it is incredibly difficult to block.

Leo: But I don't want a cookie relationship with DoubleClick.

Steve: With DoubleClick. I think, well, so here's the real concern. First of all, if you - and I did this this morning. I started off clear with my browser, fired it up. I logged onto PayPal. That is, I went to www.paypal.com. I hadn't even at that point identified myself. And I got this big, this happy screen that said, "Ready, set, shop. PayPal Plus." This was offering me the PayPal Plus credit card.

Leo: Yeah, which I always say no to.

Steve: Well, there was a big happy orange button there that said "Apply Now." I hover the mouse over that and look down at the status line of the browser, using Firefox. And I see https://ad.doubleclick.net/clk: and then a big number, 118531265, then another semicolon. Sorry, the first one was a semicolon. Second semicolon and then 11466062, another semicolon, and then a bunch of other mumbo jumbo, then a question mark followed by https://www.paypal.com/us/cgi-bin blah blah blah. Essentially, the button that you press saying I want to apply for a PayPal Plus credit card is actually a DoubleClick URL. The URL you want, that you actually want to get to, that is, the PayPal URL, if you wanted to apply for this card, that's been added to the end, just like we were saying before, as data to that DoubleClick.net URL. The problem with this is that this then creates a first-party relationship between your browser and DoubleClick.

Leo: Well, you've gone to the DoubleClick site.

Steve: Yes. You are explicitly saying I want to go to DoubleClick. So whereas...

Leo: You don't know you're saying that. That's the problem we have with this.

Steve: Well, yeah. I mean, you have to be a sophisticated user. You have to hover your mouse. You have to look down at the status line. You have to know what all that mumbo jumbo gobbledygook means. And so essentially - so you are clicking a DoubleClick URL. So that if you have - you're a person who has expressed their desire not to be tracked, not to have third-party relationships, you've disabled that in your browser, this specifically circumvents that. Now, the other worrying thing is these numbers which have been tacked on here. I watched them, and it's very clear to me they identify me.

Leo: Oh, boy.

Steve: That is, this is...

Leo: Is it always the same when you do it?

Steve: Yes.

Leo: Ugh. Oh, that's awful.

Steve: So what this is saying is, that is, I brought up a custom page at PayPal. PayPal has on-the-fly designed these links so that there is information about me that is in the URL going to DoubleClick.

Leo: PayPal knows who you are, obviously. I think they know a lot about you.

Steve: Yes. They've got my bank account number. They've got my credit card number. They've got, you know, I've got little SecurID tokens, I mean, you know, the VIP PIP tokens. I mean, I've got an extensive relationship with PayPal which is apparently now being shared deliberately with DoubleClick. And so there's one thing to know here. First of all, you could argue, you could forgive a website that wanted to get revenue from having a DoubleClick.net ad on their page. Yu could forgive them the fact that DoubleClick is using third-party cookies to track people who go to their site. It's sort of like, okay, well, cookies weren't meant to allow that, but they do. But all browsers one way or another have a way of turning that off for security and privacy-minded people who don't want that kind of relationship. This circumvents that. This creates a first-party query to DoubleClick with the actual URL you want to go to on the end. So what DoubleClick does is after they capture your identity and the cookie, which your server will be sending back, which cannot be blocked through any third-party mechanism, then they send you to the PayPal page.

Leo: Oh, so you go back to the PayPal page.

Steve: Yes, because remember, the end of the...

Leo: You're still getting it from PayPal.

Steve: Well, you go to DoubleClick. And after that question...

Leo: It says hi, I know you, I have a relationship with you. And now let's go back to PayPal.

Steve: Yes. And now we're going to return you to your regularly scheduled page. And so the URL tail is the PayPal URL that that button should have pointed to, but you were taken through DoubleClick in the process. And it looks to me like this is a static PayPal account ID given to DoubleClick, identifying me. Now, you know, it doesn't have my name. However, we don't know what sort of a relationship, we don't know how deep this relationship goes. It could very well be that DoubleClick has backend access to PayPal's database or vice versa, and there's some sort of information sharing going on.

Leo: Well, now, here's the question, is when do you get this? I mean, I guess if you're clicking on what is clearly an ad for a PayPal MasterCard, maybe that makes sense.

Steve: Ah, and I'm glad you asked the question. Because I then logged in to my PayPal account. And most of the links on the left-hand side of the page were similarly embellished.

Leo: Even non-paid links.

Steve: Yes. For example, at the top was "Enhance Your Account." And so here again I was being offered the PayPal Plus credit card. Then there was an ad with - and this was - I got a kick out of this. Unfortunately it was for the PayPal Security Key. So that also runs through DoubleClick.net. But worse, the very last link in the column was PayPal's policy updates dated August 30, 2007. The link I click on for PayPal's policy updates...

Leo: Oh, yeah, ad.doubleclick.net.

Steve: Yes.

Leo: I'm looking at it right now.

Steve: So absolutely non-advertising related. I mean, even privacy policy issues, you bend over and you are routed through DoubleClick in order to get to the page you actually want to with some sort of account ID information. Again, this is opaque. We don't know what it means. But here is my point. There is only one reason to do this. I mean, there's no reason that these links should be taking me out of PayPal to a known advertising aggregating service with opaque data added to it and then making it all transparent so that all I see is my browser landing back at the PayPal page that I was clicking.

Leo: You could easily get around this by by-hand copying and pasting the URL and then just taking the end of it. And I'm sure somebody could write, and probably will now that you've brought this up, write a Greasemonkey script of some other extension from Firefox that would strip off these DoubleClick links and go right to the redirect. I can understand if it's an ad. But when you, I mean, look. If it's a legitimate thing, I want to read the policy updates, this has nothing to do with an ad, it's routing me through DoubleClick for no reason.

Steve: I don't want to be tracked while I look at PayPal's policy updates. Unless it says, oh, and by the way, we reserve the right to share your account information with anyone we choose.

Leo: Well, it must say that somewhere.

Steve: It probably does in the fine print.

Leo: Trusted third parties.

Steve: Oh, exactly. Approved and screened, trusted third parties.

Leo: Yeah, we trust them. Now, I'm looking at all the links on here. And it is true that almost all the other links are just PayPal direct links.

Steve: Correct. They do keep you within the site.

Leo: It's the ad links. But the one that's troubling is this policy updates. Everything else, I mean, yeah, if you want to get recommended steps for merchants, that's an ad. Exchange...

Steve: Well, okay, how about - I love this one, too. "Free alerts help protect you from ID theft."

Leo: Well, yeah, that shouldn't be an ad. Oh, but it is an ad.

Steve: And that takes you through DoubleClick.

Leo: Right.

Steve: Yeah, protect yourself from ID theft by going to DoubleClick.

Leo: Well, now, we haven't contacted PayPal to get a response from them. I mean, we don't need a response. It's obvious what they're doing.

Steve: Well, and I'll tell you, Leo, I mean, I am, after everything is said and done, and I've just here been ranting for three quarters of an hour about the technology that's used, I use PayPal. I don't know of any company that is more ripe for competition than PayPal. I mean, I love the idea that I am aggregating my web purchases through them. PayPal is number one in this segment. Of course, you know, Google...

Leo: Well, here's the irony of this. I mean, Google has a merchant's card, merchant service, PayPal kind of service.

Steve: Here it comes. And...

Leo: But Google owns DoubleClick. So Google doesn't need to route it through DoubleClick. So, you know, this could be happening with everything you do. In fact, it is, in effect, because I don't know what Google shares with their subsidiary, DoubleClick. But presumably they share everything with them.

Steve: Whatever they want to, certainly. So you could argue that the fact that this is - at least it's transparent on the PayPal site. And as you said, it would be possible to write some sort of a page-scraping system that would remove these DoubleClick references and replace them with the actual PayPal URL.

Leo: It's to do, actually. Yeah, you just strip off everything up to the question mark.

Steve: Yeah. The only thing, I mean, okay. So I wanted to just get on the record that I use PayPal. I'm happy that I've got, I mean, I'm not a...

Leo: We use PayPal. We actually are actively encouraging people to use PayPal to donate to the podcasts. It drives me - this drives me crazy.

[Talking simultaneously]

Leo: ...clearly an unsafe situation.

Steve: Well, unsavory, at least.

Leo: Unsavory.

Steve: Yeah. I mean, so we need more competition in this space. We need - there's clearly a need. I mean, we've talked about PayPal before. I mean, I'm happy that we've got the really cool VeriSign dongle. I use it. I love it. I now have associated it also with my credit card form factor that's in my wallet. So I'm no longer in a situation that I used to be of my little football being at home, but I'm at Starbucks with my laptop, and I want to buy something, now I can do that. And I love the idea that while I'm - unfortunately, I need to trust PayPal. Hard as it is, I would rather they kept all my credit card information when I'm going to random, small, one-off websites I'm never going to go to again. I do not want to create an account. I do not want to share my credit card information with them. And so I swallow, and I use PayPal because it's better than doing that.

But, you know, again, this all feels like early stages of the development of the future of eCommerce. And at the moment PayPal is the organization that we are pretty much forced to deal with because that's the option we get when we go to a site is, oh, would you like to buy with PayPal. Yes, much more than giving you my credit card information. Horrible as PayPal is in these - as much as this raises a concern. The problem is, why do these links go through DoubleClick? Why is there...

Leo: What possible reason could there be?

Steve: Yes. Exactly.

Leo: Except to know what we're doing and what we buy and - because once we've established - now, here's the question. Once we've established a relationship like this with PayPal - I mean with DoubleClick - they're not - they can't track us within PayPal as we buy things, for instance.

Steve: Well, we don't know, as I said, we don't know the nature of their behind-the-scenes database connection. I mean, who knows what sort of money they're raising from each other by what kind of information they've decided they're going to share. I mean, it could be anything. PayPal has a complete chronology of everything I have purchased because, again, this is the danger of a third-party aggregator, and PayPal is one. You know, it's like, oh, click the history button, here's all your account activity. So they know what I'm - how much I'm paying for what, and where I'm buying my stuff from. PayPal knows that. So here we've got this...

Leo: Just to look at PayPal's privacy statement, they essentially say at the beginning of the privacy statement that they collect everything. Of course they do. Including credit card information, home addresses and stuff. So they know everything. We may share your personal information with members of our corporate family, which include eBay, Shopping.com, and Skype; service providers under contract who help us with parts of our business operation...

Steve: Gee, I would say that having a URL on the first page of PayPal, I guess that qualifies it as being helped.

Leo: They do say our contracts dictate that these service providers only use your information in connection with the services they perform for us and not for their own benefit. That's good. Financial institutions we partner with to jointly create and offer a product like the PayPal - oh, here we go, now they're talking about DoubleClick - such as the PayPal Plus credit card. Oh, no, that says where we share information with GE Money Bank. That's for the approval. Credit bureaus, companies we plan to merge with or be acquired by. Law enforcement and other third parties...

Steve: Look at all the data we have that you'll be able to get when you buy us.

Leo: They certainly have a lot of information. Now, you can - it says you can restrict PayPal from sharing your personal information, some types. I don't know, I'm going to go look and see what kinds of things. It doesn't sound like I have much control over this.

Steve: Well, I want to absolutely extend an invitation to anyone from PayPal to explain this. I mean, we understand the technology. This really looks bad. And I know that our listeners are passionately interested in this issue because they've been bugging me to spend the time to talk about it. And now we have.

Leo: Well, it also shakes the foundations of our whole trust system because TRUSTe is their trusty licensee. And TRUSTe validates their privacy statement. Which means to me, well, if TRUSTe's validating this, and somehow within this legalese they're saying we can do this...

Steve: Well, but Leo...

Leo: ...we're not safe anywhere.

Steve: What you just read does give them the permission to share whatever they want to with DoubleClick.

Leo: Well, it's worded, though, in such a way that DoubleClick can only use it in ways that PayPal says they can use it.

Steve: Well, no, in ways that are in line with the relationship.

Leo: Services they perform for us and not for their own benefit.

Steve: Yeah, okay, well, so we don't know what those are.

Leo: Right.

Steve: Anyway, I want to say to anyone at PayPal, we want to keep this balanced. If there's a way to explain why this is being done, why PayPal links take me through DoubleClick's server with this account number-looking information, and only then do we go back to PayPal, what possible purpose does this have other than violating our privacy.

Leo: Right. Well, I think that's pretty clear.

Steve: Yeah, I'm just saying, you know...

Leo: Maybe there is a bond there could be, who knows.

Steve: The technology is very clear. I just don't know how this could be defended. Just it's really sad to see. And again, as I said, I can't imagine a company more in desperate need of competition than PayPal.

Leo: Absolutely. Absolutely. Well, you've raised such an interesting issue. And it scares and frankly depresses me because we are so reliant on them for what we do. I don't know, I mean, I guess I could start using another merchant account system, but everybody uses PayPal.

Steve: Well, and as I said, the obscure sites I go to that I don't want to trust with my credit card information...

Leo: You trust PayPal.

Steve: Yeah. Yeah, exactly. Oh, and I have tried, as I mentioned once before, this virtual debit card, looks like a fantastic system, where you download a little applet on your machine. And in fact - oh, there's another problem. The download link for that applet is a DoubleClick link.

Leo: Yeah, see, that baffles me, too. What, you know, what possible rationale could they have for that?

Steve: I know. But my point was that due to something squirrelly in my own personal credit reports that are held by the three main credit clearinghouses, there's something that prevented me from getting automated approval. This whole thing is supposed to take 20 seconds, and you automate it, I mean, because I would love to be using this virtual debt card. This issues you a temporary one-use credit card number on the fly to use in these sites that don't support PayPal, where you don't want to give them your real credit card. I would love - I've tried three times to get this. But something squirrelly in my own credit report prevents the automated system from functioning. And I have been on hold for hours. I have tried over and over and over. I've never been able to talk to a human being to get this thing resolved. So it's like, okay, I just - for whatever reason, I can't have one of those, much as I would like to. So, I mean, PayPal is difficult to actually talk to. They're an automated faade. And people who breathe just - they don't seem to have much interest in talking to us.

Leo: Well, that's pretty much the state of the union anyway, isn't it? After all, that's where everything's headed. But we're going to fight against it. We're going to fight for your privacy and fight for companies that will actually talk to you. And people can vote with their feet. Although, you know, our server company takes its payment directly through PayPal. Many of our providers, our web designers, take their payment through PayPal. I can't - I couldn't get out of PayPal if I wanted to at this point.

Steve: Exactly. I mean, they're in a position where they have this much power. And I guess what we're seeing now is them really not caring.

Leo: Well, the word goes out. Let us know, PayPal. The nerdy Steve Gibson can be found at GRC.com. That's where Steve lives. His site is everything, everything's there. ShieldsUP, all his free security programs, the new PPP software. I just forwarded you a message, somebody rewrote it for Perl.

Steve: Yup, got it yesterday, Leo, thank you.

Leo: That is so cool. You can also, by the way, get ShieldsUP, try ShieldsUP there. That's the program that lets you test your firewall or your router. And of course who could forget SpinRite, the ultimate disk recovery and maintenance utility. It's all at GRC.com, along with this podcast in 16KB versions, transcripts by the great Elaine, who's working on Thanksgiving Day. We didn't mention that, by the way. Happy Thanksgiving for our U.S. listeners. Turkey Day today. Actually we're recording this ahead of time. What do you do for Thanksgiving? You going to have a turkey?

Steve: I would normally - I used to travel up north to visit my family in Northern California. But after 9/11, you know, travel just became so, I mean, it was already the busiest and most annoying travel day of the year. But it just became too tough. So I hang out with a bunch of friends who are also sans family for Thanksgiving, and we make up our own little virtual family.

Leo: That sounds great. That sounds wonderful. Well, have a great Thanksgiving, Steve, and have a happy Turkey Day, everyone. Or most likely you're listening to this after your turkey. Well, maybe not.

[Talking simultaneously]

Leo: If you're still listening after your turkey, you must have had some coffee, too.

Steve: Dedicated nerds.

Leo: Dedicated nerds. Some people, you know, after the big meal they go watch the football game. The real nerds listen to Security Now!. We'll be back next week. We'll have your questions and answers for Episode 120. And I will see you then. Thank you, Steve.

Steve: Thanks, Leo.

Leo: Have a great day, a great weekend. Happy Thanksgiving.