DMARC - eMail Security

Description: After catching up with the week's news, Steve and Leo look at the state of the slow but sure and steady progress being made to tighten up the Internet's eMail security. Since spoofing and phishing continue to be huge problems, these problems continue to command the attention of the Internet's largest commerce, financial, and social networking domains. The good news is: There's good reason for hope!!

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-353.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-353-lq.mp3

SHOW TEASE: It's time for Security Now!. What if spam went away entirely? Well, it seems to be a problem of authentication. Steve's going to take a look at things we've tried so far and a new technology, DMARC, that promises to fix the spam problem. That's next on Security Now!

Leo Laporte: This is Security Now! with Steve Gibson, Episode 353, recorded May 16th, 2012: DMARC Email Security.

It's time for Security Now!, the show that covers and protects you everywhere you go online, thanks to this man here, our Explainer in Chief, Mr. Steve Gibson of GRC.com, the Gibson Research Corporation. Steve is a security expert and also the author of a great utility called SpinRite that everybody needs for hard drive maintenance and recovery. Hello, Steve Gibson.

Steve Gibson: Hey, Leo. Well, 353 podcasts as of today.

Leo: Add two, though, because we did two off the books.

Steve: We did, indeed. Anybody who is waiting for Part 2 of The Sugar Hill can find it over on the TWiT Specials page, which we did last Sunday. And I'll just mention that the pages at GRC regarding health, which are under our main menu under Research, have matured a great deal. I'm getting - and Leo, this is for you, too - some fantastic case histories which I am able to share with their authors' permission in every case. I make sure that they don't mind. And they're anonymous anyway. But really interesting experiences that a lot of our listeners have had. Many of the questions they've had have been answered by these two podcasts. And in some cases they'd tried it and been

discouraged, but now they're encouraged to give another shot, or they learned enough that they think maybe they've got a better grip on it now. So anyway, I would commend anyone who's interested to go look at GRC.com/health and look at low-carb stuff. And there's a bunch of pages, all of the book recommendations. I have a Q&A page, a users' experiences page, and a bunch of resources there.

Leo: It's very timely because you know HBO's doing a four-part special right now called "The Weight of the Nation." It's all about how fat we are. There is an obesity epidemic, I mean, a huge obesity epidemic that just occurred in the most recent couple of decades. Now 37 percent of America is obese. Not just overweight, but morbidly so. So it's a very interesting thing.

And given the information that you've given us in the last two episodes - and, by the way, it's on TWiT Specials, TWiT.tv/specials, Episodes 124 and 125 of our TWiT Specials. It's interesting to watch these shows because there's still a considerable amount of confusion. I was talking to my physician about this today, actually. And he pointed out that doctors, MDs, are not trained in nutrition. They know scant information about it unless they've studied it themselves in some other way. There's a lot of misinformation from epidemiological studies, which we've talked about before, which have the basic fundamental problem of matching causation to correlation, which of course, as we know, correlation is not causation because we're all scientific thinkers here. And he said the real problem is you can't really do long-term, double-blind tests on nutrition.

Steve: Right.

Leo: You need 10, 20 years to know if this - because it doesn't kill you, none of this kills you right away.

Steve: Exactly. And not everyone responds identically. You can feed two people the same diet, and they will get completely different responses.

Leo: He said the same thing. He said it's kind of you can't be sure of what the effect will be. So you have to do what you're doing, which is do a lot of research and try to find something that does work for your body. And you're doing this, I should say, under medical supervision.

Steve: Yes. And I test weekly. I've got weekly charts of all of my...

Leo: You're taking this very seriously, yeah.

Steve: ...yeah, of all my measurements. It's interesting. Currently the bottom testimonial, such as it is, or it's feedback, I got permission to post just this morning, so I put it up. It's a really nicely written piece about how a guy went low-carb and reversed his diabetes and high blood sugar, brought his blood pressure down, his HDL (the good cholesterol) up, the LDL (the bad cholesterol) down, triglycerides way down, I mean, everything was perfect. He goes to his doctor, and the doctor says, "My god,

congratulations, this is fantastic. You're going to live forever. What are you doing?" And he says, "Atkins." And the doctor goes, "Aghhhh."

Leo: Oh, yeah, right, that's terrible for you. Don't you know?

Steve: "It will ruin your liver. You're going to ruin your kidneys." And so the guy says, "Well, how do my liver enzymes look?" And the doctor says, "Well, they look perfect, they're fantastic. But you're going to ruin your liver, and you're going to ruin your kidneys. And you're going against the USDA and the FDA," blah blah blah. And then he says, "Well, how does my urinalysis look that would show what's going on with my kidneys?" "Oh, it's perfect. There's nothing wrong. But you're going to ruin your liver. You're going to ruin your kidneys." And this, of course, is - it's the truth. And so, you know, he was discouraged and thought, well, god, I mean, it worried him that somebody who he assumed was an authority would have this opinion. And so my favorite set of books are those low-carb - "The Art and Science of Low Carb Living" and "Low Carb..."

Leo: I ordered those.

Steve: Good, because these guys are double PhDs and MDs who have been studying this. And if nothing else, they give you some objectivity that I don't think that we're finding right now. And so anyway, it's interesting stuff, I think. It's fascinating to me.

Leo: You know what one of the conclusions of this "Weight of the Nation" is the government needs to regulate all this. And I thought, you know what, the government has mishandled this from day one.

Steve: Yes.

Leo: And that's part of the reason we're in this fix. Maybe if we just - if people just studied and learned and thought and did what you do, which is do this under doctor supervision, I think that you can do it yourself.

Steve: Yeah, and experiment. I think it might have been in the beginning of "Good Calories, Bad Calories" that Gary explains how political this process is to government.

Leo: Yeah, alas, yeah.

Steve: That in fact it's not, I mean, the scientists are consulted and then ignored because the government and the political advisors don't like what the scientists say. So this isn't science-based. It's politically driven, in fact.

Leo: Yes, as is most politics. Most government is politically driven. Anyway, enough of that. Let's get into the security issues. Today we're going to talk about what?

What is DMARC?

Steve: I want to talk about what's been going on quietly and in the background by the big players in the industry that are essentially the movers of this towards dealing with the problem of phishing and spam and email spoofing. Everyone's just sort of given up at this point. And we know that there's still a huge problem because, in the same way that I'm noticing sites are now saying you need to have JavaScript turned on, because I'm going there initially with it off, and because for safety, and then I turn it on when I need it, similarly - and I'm sure you're seeing this, too, Leo - there are more and more sites that, when you do something with them, and they need to send you a confirming email, they say, you know, warning, if you do not receive email from us within two minutes, please go look for it in your spam folder because we really have sent it, and we're really going to, and you need to receive it in order to verify your email address, and it may have trouble getting to you.

And so this represents a big problem for the Internet. We've had problems with email from the beginning. It is still the No. 1 means for these large site takeovers. We know, for example, that that whole RSA fiasco with them losing the keys began with an employee clicking a link in spoofed email that then allowed an exploit to take hold that allowed the bad guys to gain entry into the RSA network. So it is really important for Internet security.

Well, there have been some efforts made that have sort of sputtered along. We're going to talk about what they are, and why they haven't been able to get the traction that they could, and how, despite the fact that we haven't been paying any attention, there is an effort ongoing which is moving forward that looks like it's going to give us, finally, some real relief. And we're about at a tipping point. I would imagine, as often is the case on this podcast, we'll be talking about it. It comes across our radar early; and then, oh, maybe six months to a year from now, suddenly it's going to be...

Leo: Everybody's talking about it, yeah.

Steve: ...oh, yeah, we already know about that. We did that in Podcast #353 a year ago.

Leo: It's funny how much people have stopped talking about spam, not because spam has gone away or is any better - in fact it's, as far as I know, worse - but our tools for fighting it have gotten better, and particularly filtering. But this goes beyond filtering. All right. Before we get into DMARC, Mr. Gibson...

Steve: Yeah, we've got just a little bit. I don't know, sometimes a week just gives us so many news topics to talk about that we barely get to the show. And in fact sometimes grumpy people will send me a tweet saying, "It took 55 minutes before we actually got into the content." It's like, hey, folks, news and security updates and conversation about the week's events was the original concept for this podcast, and we added all of that other content afterwards. So, I mean, I spend as much time, well, not as much time, but lots of time wanting to make sure I keep everybody current with things that are going on.

Leo: But see, that's why this show could be any length at all. You get to choose the length because it expands to fit the needs.

Steve: Oh, and I also get tweets saying, "Love the podcast. Could you do it daily?"

Leo: [Laughing] No.

Steve: Just scrape me off the floor.

Leo: I can tell you right now, no. But weekly's good. Weekly's good. You should be - weekly's good.

Steve: It's just right.

Leo: Yes.

Steve: So Chrome has moved itself to v19. And in the blog yesterday, quoting from the blog, they said, "Say you've found an awesome recipe on your work computer while, ahem, working hard at the office. But when you get back home, you can't quite remember if it was two teaspoons of baking soda or two teaspoons of baking powder."

Leo: [Laughing] Big difference.

Steve: I'll take your word for that, Leo. "Wouldn't it be cool..."

Leo: Yeah, you're such a cook.

Steve: Yeah. "Wouldn't it be cool if you could pull up the same recipe on your home computer with one click? With today's stable release of Chrome, you can. When you're signed in to Chrome" - and we'll explain what that is in a second - "your open tabs are synced across all your devices, so you can quickly access them from the Other Devices menu" - which is a new menu - "on the New Tab page. If you've got Chrome for Android Beta, you can open the same recipe tab right on your phone when you run out to the store for more ingredients. The back and forward buttons will even work, so you can pick up browsing right where you left off." So that means they are even syncing the history of the tab that is now being bridged by these devices.

And so then I thought, well, okay, what does "signing into Chrome" mean? And they said: "Signing into Chrome brings your bookmarks, apps, history, and other settings to all your devices. Anything you update on one device instantly updates everywhere else, and your Chrome stuff is safe in case anything happens to your computer. It's your web. Take it with you." Of course, and Google, too.

Leo: Very nice.

Steve: But, so, yeah. I'm wishing that Chrome wasn't just becoming so neat-looking because...

Leo: Ah-hah. You're feeling bad now, huh?

Steve: Well, I'm feeling torn. I'm feeling pressure because I'm so happy with Firefox.

Leo: But you're a heavy tab user.

Steve: Oh, boy, I am. And that's the problem is I've got 48 open right now.

Leo: Oh, my god.

Steve: And side tabs were taken away, as we talked about before, because they decided, well, it was just experimental. And people screamed so much they said, don't worry, we understand there were people who organized their lives with tabs. As I do. It's just my - it's my things to come back to later list. But I still like the way Chrome manages memory. When you close a tab, like I watch my memory, and it's like, bang, memory is returned to me because Chrome runs every single page in its own separate process as part of their inter-page security isolation, which is different, entirely different than what Mozilla does with Firefox. There's just one Firefox.exe. But when you've got Chrome running, there's about seven Chrome EXEs with additional ones for more pages because of the way they've set up their architecture. And then those, of course, communicate in a safe way.

So, yeah, I mean, Google, I love it that there's a strong commercial drive behind Google's wanting to keep making Chrome better and better, and that Google's doing it. Microsoft just seems to have stalled in innovation land. And so, like, they're not using...

Leo: Oh, yeah, IE's not - I don't know where IE's going, yeah.

Steve: Yeah. I got a tweet from an old friend of ours, Alex Neihaus.

Leo: Oh.

Steve: Remember Alex?

Leo: Yeah.

Steve: He was - was it Astaro?

Leo: Yeah, he was at Astaro.

Steve: Yeah, in the early days.

Leo: He was the guy, he was our first - got our first ads on the whole network, which was Astaro for Security Now!.

Steve: Yeah. He tweeted, and I just wanted to share this to remind listeners. He said: "@SGgrc, web.tweetdeck.com now turns on both SSL and SPDY indicators in Chrome on my Mac. It's beautiful and fast." So that means that Twitter has it at their end, and TweetDeck has it at its end. And I just want to remind people, I'm a TweetDeck diehard because I'm so - I just have it open on a monitor, watching with, like, six columns open, keeping an eye on things and often replying to people who tweet.

I do need to remind people that I only am willing to do a DM. I can't, I don't want to clutter up my feed with lots of @ mentions. So if somebody wants a reply from me, you've got to follow me. And then if it's a tweet that I have time for and can and want to reply to, I'll do that with a direct message. But I can't direct-message people who aren't following me. And that's a good thing for keeping Twitter spam down. But often I'll see something, I'll go, oh, and I'm excited to respond. I do a direct message, and it says "Recipient not following." It's like, okay, well. So I can't reply when I otherwise would have. Or, I mean, when I did, and it just got blocked. So for anyone who wonders why they never heard back, that could be why.

And I just have, before we get into our content, another interesting SpinRite tip, different from what we've talked about before, from a Sam Fineberg in Palo Alto who wrote on April 26th, so just last month. He said, "Two of the computers in our household, my son and wife's, were running really slow. My son's computer was even overheating and shutting off occasionally, primarily when he played Minecraft. The computers weren't that old, so I was contemplating reimaging them. In preparation for that, I ran a fresh backup on my son's computer and decided to run SpinRite on the drive.

"As I was running SpinRite, I noticed the drive got very hot, and there were millions of seek and ECC-correctable errors. Running on a similar computer that wasn't slow, there were very few seek errors, and the drive stayed much cooler. So I did an image copy of his drive to a new one, installed it, and the new drive is working great, faster, and no overheating issues. After seeing that, I ran SpinRite on my wife's computer. It also had millions of seek errors. Once again, I replaced the drive, and it solved that computer's problems, too. So in this case, SpinRite wasn't needed to fix a problem, but it led me straight to the right solution."

And what was happening there is, this is, again, one of the things I mentioned last week. I talked about how subtle problems with cabling could be inducing errors which SpinRite shows you when you're running it in that cabling errors count. Similarly, unlike any other software on the planet, there's nothing else that I've ever seen that shows you the error corrections that are actually being done on the fly, which are hidden from you, and also problems that the drive has finding the sectors. The reason there were all these overheating problems is that the heads were servoing a lot, and that's where you use up a lot of energy.

So, again, SpinRite, on its statistics page, it's actually on the SMART, the self-monitoring and advanced reporting technology page, where it's dynamically looking inside the drive and bringing this real-time data out. And what Sam did was exactly right. He looked at a similar machine with, like, the same make and model drive, and saw what it was doing. And that gave him a baseline to compare the other drives, which were clearly misbehaving. And so essentially they were at some point going to fail, and SpinRite was saying, look at what's going on here. This is important. And so he was able to swap them out for replacements that just don't have that problem. So another aspect of SpinRite, really the preventative maintenance and monitoring side, which can really come in handy.

Leo: Really neat. Let's see, here. I think we want to get to the matter at hand.

Steve: Yeah.

Leo: And then I'll interrupt you in a few minutes.

Steve: Okay.

Leo: Because I don't want to do another ad right away.

Steve: Perfect. So DMARC is a somewhat awkward acronym. I don't know how the military gets these amazingly cool acronyms, Leo. But somehow they always just have these, like, okay, how did you get that?

Leo: It's a retronym, where they start with the name, and then they make up what the letters do maybe, I don't know.

Steve: Yeah. And this sort of feels that way, too, like demarcation, demarc, dmarc. But this is kind of awkward: Domain-based Message Authentication, Reporting, and Conformance. It's like, eh, okay.

So, okay. So here's what's going on. There have been, for actually a decade, if you can believe it, a couple specifications that were supposed to solve the problem of spoofed email. We've talked many times, in many different contexts, about email spoofing, that it's a big problem because the headers in email can be set by the sender, so it takes a real expert to examine the headers. And few users are. And in fact, if a piece of email bounces around a lot, each bounce appends headers onto the beginning. So the order of the headers is in the reverse order of the path the email took. And it ends up being complex. And, boy, headers these days are just so full of gobbledygook, it makes your eyes cross to look at it. It's not like our grandparents' headers.

So the effort has been to come up with a way of preventing spoofing. So one approach was the so-called Sender Policy Framework, SPF, which is 10 years old, which surprised me, just didn't feel like it had been around that long. But it's a clever solution for, again, leveraging DNS. One of the things we're seeing is other new applications for the domain

name system, where new record types, or repurposing of existing record types, which is what we do here because adding new record types to DNS requires multiyear standards, and old servers won't support the new types, and so that's going to be a huge drag for something not DNS-related.

But reusing existing record types, for example, the text record, which is what these systems use, that's easy. You don't have to change anything because all DNS servers have long understood what a text record is. It's just sort of a freeform text record. You can have as many of them as you want. And you could say, "Give me the text records for Google.com," and it sends them to you. So this is another reason why layering security on DNS is a good thing because this domain name system is increasingly going to become the master lookup index, domain-based lookup of information for the 'Net. Traditionally, all you could look up was IP addresses based on a domain name, or the reverse of that, domain names from an IP address. But as the 'Net matures, we're seeing new applications for this same - the same system.

In fact, GRC uses it to distribute the version numbers of our software. You're able to say application.dns or .version.grc.com, and the IP address that it returns is the latest major and minor version number of our software. So I do that because it generally passes through firewalls. It's super lightweight, just a UDP packet out and a UDP packet back. And it's a nice little way of getting some information from GRC.

So there are increasing number of things that use DNS. And so securing that translates into all kinds of benefits as we come up with additional uses for it. So what SPF, the decade-old Sender Policy Framework does, is it's a way for the owner of the domain - and that's the other kind of key concept here is the notion of who controls the information. The reason we can rely on DNS, for example, to get the correct IPs for Google, is that Google controls the Google.com domain. And all of the enforcements are in place for that to happen. I control my own DNS server, or servers, that then supply the data records to Level 3, that provides me with my connectivity. And so people actually ask the Level 3 servers for GRC information, which they get behind the scenes from me. But the point is that I can have that information say anything I want.

So when SPF came out, I immediately adopted it. I added records because it took, like, five minutes, was easy to do. The idea is the recipient of a piece of email - and I don't mean the end-user recipient, but the SMTP server that the end-user is using. So, for example, if you were a Gmail user, then it would be the mail.google.com server. It's the server that we connect to as clients to collect our mail, whether it's IMAP or POP protocol. That server, when it's having a conversation, receiving email, the email says I'm from Google.com.

Well, the receiving server is able to, right then, do a DNS query for the text records for Google.com. And among them will be an SPF record which specifies the range of valid IPs or an enumerated list of IPs. There are a number of formats. Or it can even give a domain name for referral, that is, the valid originators of this email. And the idea being that then this receiving SMTP server can check the IP that this connection is from. And we know that IPs cannot be spoofed, IP addresses, because they are point-to-point links. And part of the TCP handshake is a multidirectional packet exchange, specifically for the purpose of verifying that packets can go between the two IPs.

So unlike UDP, that is one-directional with no verification, TCP has verification. So the receiving server, the server receiving the email, absolutely knows the IP that it's connected to. And if it receives that IP from that domain's SPF record, its declaration of what machine's IPs are valid senders, then it's able to verify that email on the spot because it knows it's connected there. The person who controlled the domain has

specified this IP is valid. And so it cannot be spoofed from someone else.

Normally, without this kind of verification, there's nothing to prevent some random server anywhere else on the planet connecting up to Google.com and sending email absolutely as if - or, I'm sorry, connecting up to your SMTP server and sending email that looks exactly like the email coming from Google and declaring that it's from Google.com because most times email servers are not the same IP as, like, the main web domain server. And there is a provision in DNS for having a mail server declared as mail.google.com. But oftentimes there's big networks; there's load balancers. All kinds of things make this more complex. So this was the very simple concept behind the old Sender Policy Framework technology.

Now, the benefit of it was that it was trivial to implement. All anybody had to do who wished to authenticate that email they were sending was add a simple text record to your DNS server. And at the SPF site there was even a little helper where you could specific the parameters that you wanted to use, like if it doesn't match, do you want me to drop the mail? Do you want me to call it spam? How seriously to take a failure to match and so forth. Like is it advisory-only, or is it serious? There are a number of parameters. So you were able to set those, press a button, and it would give you, it would build for you a little SPF record that you could just then drop into your DNS server records.

The problem with that is that, traditionally, and that's certainly more the case 10 years ago than now, email was a store-and-forward technology. And that store-and-forwardness was a source of great abuse. But that was the original concept of email, the idea being very similar to the way we have packet-switching on the Internet. We've talked about that often, where you just drop a packet randomly onto the 'Net anywhere with an IP address, and the job of routers is to bounce it from one router to the next, always aiming it towards its destination. Similarly, email, sort of operating - instead of operating at the link layer there, operating at the application layer, email was the same way. You might have email bounce a couple times between SMTP servers. There were servers that would accept email from anyone and then initiate a connection and try to forward it, a so-called "store-and-forward."

And what happened was there were many servers that were open servers, the "open relays" is what they were called. And this whole notion of store-and-forward was like relaying email from one server to the next. And in the old days, before spam was a problem, typically servers were all like this. They would happily relay email on behalf of someone else because once upon a time we were all good guys, and there were no creepy bad people hanging out on the Internet.

So that became abused immediately. One of the first tricks of the spammers is they would find a so-called "open relay," and they would dump their spam there. If their own servers had been blacklisted by their IP address so that nobody would accept any email from them anymore, they said, okay, fine. So they would just search around the Internet for so-called "open relays," and they would dump their email on that machine. And then it, doing its job, would forward it to its destination, which was typically a spam recipient, who wasn't happy. And of course you then, "you" the open relay, would end up getting blacklisted because now your IP was generating spam, even though you yourself weren't generating spam.

So over the years this got shut down. It's very rare now, and it's considered an SMTP email server configuration failure if you have an open relay. It's like, oh, my god. I mean, and actually you find out about it pretty quickly because there are bots and things that are roaming around looking for them. And you'll generally get, if you've got like an admin

account on your email server or something, you'll get email saying, "Warning: You are running an open relay." And oftentimes you'll be getting spam complaints and all kinds of problems. So that's been bolted down.

So the original problem with the Sender Policy Framework is, when you think about it, is it was completely incompatible with mail forwarding. The recipient is verifying the connection IP address of the sender domain. So that implies it has to be point to point, that is, it has to be from Google as the originator to your server, and for your server to look at the IP addresses that Google is advertising through its SPF record as valid and say, yes, we believe this is Google. If there were a bounce in between, if there was any forwarding going on to some third-party SMTP server, well, the Sender Policy Framework, with filtering, would reject it because the connection IP would be other than what Google is advertising.

So 10 years ago this was a problem with SPF. It's not a problem today because nobody's relaying email anymore. We've got robust networks. We've got super inexpensive hardware. We've got load balancing and all these other solutions to the original problem of, like, someone's server being down, so you'd send the mail and park it somewhere until their email server came back up, and then you'd be able to get it over. I mean, so the relaying is shut down, and suddenly SPF has a real chance to work.

Now, there was a second effort, which Google adopted early, which was called DKIM, which is an acronym stands for Domain Keys Identified Mail. And it's a somewhat complex specification. But what it is, is pretty simple. Listeners to the podcast will understand it immediately. It is simply using public key crypto to digitally sign all outgoing email. So anything originating from a DKIM-signing SMTP server is signed using that server's private key. So, and this is compatible with, like, other signing. For example, you could be using GPG, PGP, whatever, to sign your own email. But then that's been wrapped in an envelope with all the headers and delivery information. Then that is signed, as the final phase, by the server sending the mail out, using the server's private key. What that means is that the recipient of the email is able to verify.

What they do, once again using DNS, is they look up the apparent sending domain's DKIM public key. So the public key is published in that domain's DNS record, once again in a text record, in a specified format. And all these have well-established specifications. And so the recipient is able to get the matching public key and use that to verify that domain's signing of the envelope of email. And since the private key is never disclosed, there's no danger. There's no way for any third party to falsify that signature. It's full-on, state-of-the-art crypto strength and solid.

So those two different solutions have existed. But they haven't gained traction, or they've been doing so very quietly and silently. One of the problems was the so-called "YAP" problem, Yet Another Protocol, where it's like overworked administrators, they hear about this, it's like, oh, boy. We have other things to worry about right now. Also there was a problem of very complex infrastructures. For example, these large ISPs have massive networks, huge blocks of IPs. They may be adding blocks of IPs for other purposes. They were worried about false-positive rejections where, if they changed some configuration so that mail, their own legitimate mail would now be routed somewhere else and come out of somewhere else, suddenly that was going to be published. DNS would need to be refreshed over time.

So there was just sort of this sense of, wait a minute. We're not sure that it's worth the trouble that it's getting us. And also note that this system inherently requires that senders and receivers collaborate together, that is, the senders are saying, I'm attesting, in the case of SPF, I'm attesting that all valid email comes from this IP. But if the

receiver doesn't check, doesn't care, then this is, I mean, it's inexpensive to make that assertion in your DNS records, but you're not going to get any benefit from it unless somebody on the other end takes the trouble to check and does the right thing with the results.

Well, very quietly, five years ago, in 2007, PayPal did pioneer this approach. They worked out a system with Yahoo! and later with Gmail to collaborate this way, and the results were extremely effective. It led to a very significant decrease in suspected fraudulent email that was purporting to be from PayPal being accepted by those receivers. So, I mean, the concept is golden. It absolutely works.

And so five years ago PayPal was having this problem that their email domain was being actively spoofed, and Yahoo! Mail users and Gmail users were having a big problem. PayPal was having a problem with those users. Now, they were having a problem, of course, with lots of other users, too. The reason that Yahoo! Mail and Gmail were singled out is they were aggregation points. There were a ton of people using Yahoo! and a ton of people using Gmail. So just by fixing those two recipients, by getting PayPal, I mean, by getting those two to agree to honor PayPal's assertions about sender policy and DKIM-signed email, all of the users of Yahoo! Mail and Gmail got fraud protection for the PayPal domain. So it worked there because we had one person with a big problem was able to convince two recipients with huge email user bases to go along with them, and it worked.

The other problem that both SPF and DKIM have had is that there hasn't been any built-in feedback. I've had SPF records at GRC for a decade. And I don't know if it's done any good. I don't know if they work. I don't know if any email has been, like, filtered out or prevented. I like the idea that, in theory, if somebody checked my SPF records and saw that only email coming from this IP was valid, then GRC.com could not be spoofed, email from me could not be spoofed. So I thought, okay, that's a good thing. But I don't know if it ever got used. I have no idea.

So there's that problem, which, again, there's enough adoption resistance and inertia that, unless there's a clear benefit, it's just not going to happen. And admins have enough emergency stuff to worry about. They're not looking for more work to do. They're just trying to stay ahead of what's going on right now.

So what is little known is that, because SPF and, to a lesser degree, DKIM are so easy to adopt - SPF just requires adding a couple records to DNS. DKIM requires upgrading your SMTP server to digitally sign outgoing email. And again, 10 years ago, as we know, public key crypto wasn't free. So there was some expense in terms of computation overhead to sending out a piece of email that was digitally - that was signed with a public key. Today that's just, you know, we've got so much computing power, how many cores does your chip have, that that's just not a problem.

So adoption has been growing. And at this point a little bit more than 50 percent of all domains are actually supporting either SPF or DKIM. So again, not near a hundred, but more than half. And in terms of mail volume, 85 percent of email has either SPF records associated with its actual domain originator, or they're digitally signed. And in fact just this morning I got some Facebook notification that someone who I know knows somebody else, I don't know what it is, but anyway - because I keep trying to turn this stuff off.

Leo: Yeah, good luck. They keep turning it on, yeah.

Steve: They do. It's so annoying. Anyway, so it's got, like, DKIM-signature, and then a whole bunch of gobbledygook in the header, which I now know is, for example, it says "a=rsa-sha256." So that says it's RSA public key crypto with the SHA256, the Secure Hash Algorithm, 256-bit signed. And then the valid domain, "d=facebookmail.com," and then a whole bunch of stuff. So, I mean, it goes on and on and on. There's, like, a block of gobbledygook. But that's all the signing of this. Oh, and I got a kick out of this. It also says "x-facebook: from zuckmail." It's like, yes, well, we know where they...

Leo: Yeah, zuckmail, I like that.

Steve: Zuckmail. So, okay. So I sort of narrowed this down to four problems. Many senders, as I've mentioned, have a complex email environment with many systems sending email, often including third-party service providers. Some large domains sub out their email handling to somebody else. So it's coming from their domain, but it's being routed through somebody else. So ensuring that every message can be authenticated using SPF or DKIM is a complex task, particularly given that these environments are in a perpetual state of flux. It's like, oh, let's switch over to this third-party company. It's like, oh, well, wait a minute, our authentication is going to break if we do that. I mean, so it's like, oh, is it worth risking that?

Also, if a domain owner sends a mix of messages, some of which can be authenticated and others that can't, then email receivers are forced to, that is, the receivers of the email are forced to discern between the legitimate messages that don't authenticate and the fraudulent messages that also don't authenticate. By nature, spam algorithms are error-prone and need to constantly evolve to respond to the changing tactics of spammers. The result is that some fraudulent messages will inevitably make their way to the end-user's inbox. In other words, it's the typical soft filter, or heuristic filter. Good messages are going to get rejected, and some bad messages aren't going to get rejected because we don't have absolute authentication today. But it looks like we're headed for that quickly, or soon.

Senders get very - and this is key. Senders get very poor feedback on their email authentication deployments, which is to say none. Unless messages bounce back to the sender, there's no way to determine how many legitimate messages are being sent that cannot be authenticated - so how would I know if somebody was rejecting my email? There's just no way to know - or even the scope of the fraudulent emails that are spoofed in the sender's domain. So, for example, again, if people are sending random email as if from GRC.com, how would I ever know that? There's no way to know. So this makes troubleshooting mail authentication issues very difficult, particularly in mail environments which are increasingly complex.

And finally, even if a sender has buttoned down their mail authentication infrastructure, and all of their legitimate messages can be authenticated, email receivers are wary about rejecting unauthenticated messages because they can't be sure there is not some stream of legitimate messages that are going unsigned. And so, again, the recipient is like, well, okay, some of these are coming in signed, but these look pretty good over here that are not signed. Maybe they're coming from a new server that Facebook set up, and they don't have authentication in place yet. So all of this just created enough uncertainty that it's just kept people from moving forward.

But the big guys - Google with Gmail, Facebook, LinkedIn, PayPal, Bank of America, American Greetings, Cloudmark, Comcast, Fidelity Investments, Microsoft with Hotmail, Return Path, and Yahoo! - are all on this now. They're all part of this DMARC effort. And

they have determined this stuff is mature enough, the specs are there, they work, the technology has been moved into servers. Remember that SPF really didn't require any on the sending end, but it certainly does require it on the receiving end. The SMTP server receiving has to have new technology for going and getting the apparent sender's SPF records and checking the IPs and checking the connection. So that's taken some time for those features to be added to standard email servers. And the same is the case with DKIM on both ends. The sending server has to digitally sign outgoing email, and the receiving server has to be able to verify the digital signature from the apparent recipient, I mean, from the apparent sender.

So we're to the point now where the software technology is in place. It's now time to solve these lingering problems. So for senders, adding this kind of robust authentication requires work, and the return on investment has been uncertain. And for the receivers, just having some authentication doesn't help much. I mean, what we need is to be able to say, okay, we believe that the domain in question is asserting that it's got control of its outgoing mail, and we're absolutely going to take action on failure. So the solution is that senders absolutely authenticate all outbound mail and assert that to receivers.

Leo: Right.

Steve: So a bunch of smart guys - oh, yes.

Leo: Let's pause.

Steve: Perfect. Perfect place.

Leo: As we explain how this solution might be implemented. How about that. It's always, you know, it often comes down to authentication, but the devil's in the details. How do we authenticate? We were talking with Bill Harris, who is a former CEO of PayPal and runs a company called Personal Capital, and we were talking about micropayments and online payments and getting rid of money. And it's really an authentication problem. How do I know you're you, and how do you know I'm me? Once we've authenticated - that's the problem with credit card fraud is weak authentication methods.

Steve: Well, and it's why I went nuts over Stina's YubiKey.

Leo: Right.

Steve: It's just like, hey, here's something to replace passwords, which is such cooler authentication.

Leo: So authentication's the solution. But how do we achieve it?

Steve: Probably the biggest thing that DMARC brings is closing the loop, providing a

means for allowing senders who wish to authenticate a means of verifying what's happening out in the world before they commit all the way. So the DMARC spec, it solidifies and unifies the existing SPF and DKIM specifications and also provides some configuration guidelines because the specs were broad and gave a lot of latitude. So DMARC is saying, look, we didn't know what we were going to need initially. Turns out this is what people use, and this is enough. So here's how we want you to configure SPF and DKIM.

So the beauty is all we're talking about is some additional technology added to our email systems, but basically built on these existing established standards. There's a new DNS resource record. And again, rather than inventing their own, which would again require all DNS servers everywhere to be updated, they're just going to add a new resource record of the TXT, a text resource record, which is just anything you want to have it say. The text record specifies the sender's policies. And they have this weird word, they use it, they call it "alignment" for some reason. Alignment types is either strict or relaxed, so how strict you want the matching to be.

Then they have a disposition for the incoming - for any incoming problems, whether the sender wants the receiver to quarantine records that don't match, to reject them outright, or just to monitor. And finally, this text record contains - and this is where the loop gets closed - the URIs, or URLs, for sending reports, both of failure and aggregate daily summaries. And the DMARC spec then defines this aggregate reporting format where it's a daily sum in an XML, so machine-readable, machine-parsable format, which is deliberately redacted for the sake of privacy. That is, they want to just accumulate statistics, not specifics about individual email.

And so the daily aggregate reports are per apparent "from sender" domain, but they do not contain delivery disposition and do not contain individual email addresses. But the idea is that they, by sending domain, they feed back the authentication results for the DKIM signature verification and the SPF IP verification, along with the successes or failures of the matching, and specify the policy action which was requested and taken. So what this means is that a company that was deciding, as all these big companies I just listed are - and we know that SPF and/or DKIM is now deployed in about 85 percent of the volume of emails on the Internet and about 50 percent of the domains on the Internet.

So the next step is to take the server technology a bit further with this DMARC spec, which, by the way, has been submitted. It was finalized just this January, a few months ago. It has been submitted to the IETF, the Internet Engineering Task Force, for ratification and publication as an Internet RFC. So it's going to be a formal specification moving forward for email on the Internet.

And so the idea would be that a company that wanted to lock down their email would upgrade their email servers to support the DMARC spec and add the DMARC text record to their DNS, which is simple to do. And, for example, initially they would say, well, monitor. Don't reject, don't quarantine, don't block, don't do anything, but we want reports So they would put their whole system in monitor mode. And then all of the recipients who also supported DMARC at the other end - remember, this is inherently, for this to work, it's got to be sender and recipient. Both have to be on the same page. They both need to be supporting this.

And it's a fail soft solution, that is, if either one doesn't, you just don't get authentication. But it's looking like email is a significant enough backbone of the Internet, and companies, especially as we're doing finance more and more with banking and PayPal and email and also social networking, and where link spoofing and phishing is a problem,

if we can fix email, this is a huge step forward for Internet security.

So the problem with deployment gets fixed by creating an ongoing feedback system where initially the system's in monitor mode. And again, on an individual by individual sender basis. So if I were to deploy it, I'd upgrade my email server. And in fact the next server that I choose, this will be a requirement for me is that it be able to support this protocol because this is clearly where we need to go. Then I would say monitor. And any email outbound from us that goes to a domain that also supports DMARC, they would be querying our DNS records, see that we support DMARC, determine what our policies are.

And, for example, initially I'd be saying just monitor and report. And so I would get back to the domain and URI that I specify, I would receive a daily report from every one of the domains that we touched that day of what they thought about our proposed authentication. And so for a huge environment like Facebook, where massive network, global, servers all over the place, they want, in order for them to confidently say "reject," before they get to that point they want to be able to monitor. They want to see that everything's working. And then, even when they are in reject mode, where they're saying to people that we were monitoring for a while, we verified that everything's working, there is no mail coming out of us which is not authenticated, there's no way for mail to get out of us without authentication being added, now we want everybody to just drop any email that is being spoofed with our domain that doesn't authenticate. But we still want reports.

And so suddenly we get something that we've never had before, which is an ongoing reporting of fraudulent email going to third-party servers which support DMARC. The third-party server would get inbound email spoofed from Facebook, for example, check with Facebook's DNS - remember that DNS is caching also. So in fact they may well - major domains like Google and Facebook and so forth, these servers would have those text records in their local caches, so no traffic overhead is being required at all.

So they would say, wait a minute, this isn't digitally signed. Facebook has said reject it if it's not. And they would aggregate the information. They would log the connecting IP of the spamming, fraudulent, spoofing server and send that to Facebook. So now Facebook begins getting daily records of the IPs that are originating spam spoofing them that they've never had before.

So this robustly authenticates email. It safely allows the system to be deployed. And again, because it's DNS, if there was ever a problem, the admins could switch back to monitoring mode instantly, essentially, or over the expiration time of the DNS records which are being used. And so it's flexible. It's under sender control. And closing the loop, generating reports, is really valuable real-time information.

So potentially, once we get there and bring up authentication to a level that we have never had before, the people behind this are hoping that this will enable new forms of communication over email because it won't be something where you're looking at anything that comes in with a great deal of skepticism. And you can imagine also that at some point servers could be, like, adding a tag, or email clients could say whether the email is authenticated or not. And if it said it was, you could trust it because it would have been securely verified. So that's where we are with this. And I'm excited about it.

Leo: Cool, yeah. Very interesting. Now, how widely adopted is this? I mean, do we have to consider that? Or is that not an issue?

Steve: Well, end-users really don't have to do anything except sit back and wait a while. It is...

Leo: But ultimately we're all users; right?

Steve: Yes. Yes. Well, the spec is in place. It's been finalized. And you can imagine, I mean, among these big players, anytime you see some weird word like "alignment types," it's like, what committee group chose that? So it's like, okay. So it's a significant accomplishment that all of these players, who are competitors with each other also, got together on the technical level and said, this is what we're all going to agree to. We can't all, I mean, we already have DKIM and SPF, which are, like, competing standards. Well, they said, instead of choosing either one or trying to amalgamate them, they said, look, everybody likes their own thing. Let's just support them both because some of them are in place, and SPF is easy to do. So we don't need to choose. What we just need to do is we need some way of believing the authentication.

Leo: And that's not just out of trust. You actually need a technology to do this.

Steve: Yeah. Exactly. Yeah. You need the technology, and you need to be able to say we're actually going to throw this away if it doesn't authenticate. And before...

Leo: And that's the key. And that's the problem with SPF and everything else and all these other authentications is, yeah, if you were willing to throw away - it has to be 100 percent or very, very widely adopted. But if my mom adopts it, I can use it. Otherwise I won't get her email; right?

Steve: No, no, no. It's your mom's ISP.

Leo: ISP, okay.

Steve: Yes. So it's SMTP server to SMTP server.

Leo: Got it, of course.

Steve: And so it's point to point.

Leo: So if all the big ISPs adopt it, then we're good.

Steve: Yeah. And it's going to happen, Leo. I mean, this is - if 85 percent of the valid email traffic on the 'Net now contains this, but no one is using it, we're just, it's like, okay, everybody is saying we're ready to go. But it's time to commit now. And I think we're just at that point. I mean, this is a big change. To be able to lock down email and require authentication, what's going to happen is the big guys are going to do it, and

then that will put serious pressure on the rest of the industry because it'll be like, wait a minute, you're not sending authenticated email. I'd rather use somebody who is because it'll become a value-add.

Leo: Or you could have a folder that says "Unauthenticated." It's like a spam folder, but maybe a step down from that.

Steve: Yes, yes. It could be, well, yes. You could have a spam folder or, that is, an unauthenticated folder. And the beauty of that is it's going to actually mean something. The spam folder gets false positives, and then your normal inbox gets false negatives.

Leo: Right.

Steve: And so the beauty of this is your normal inbox would never receive an unauthenticated email. And so, again, having a sender who knew their email was going into people's unauthenticated folder is going to move heaven and earth to add authentication to their servers so that their email doesn't look second-class. It's a little bit like the extended validation certificates. GRC is now all EV. And every time I fire up GRC in a browser, and it comes up with EV, I'm like, oh, yeah, cool, I have that now.

Leo: I hope this works. We'll see.

Steve: Yeah. Well, I mean...

Leo: It's the best shot yet.

Steve: And this just shows how much inertia there is. I mean, the system isn't so badly broken that no one uses it. It's just so badly broken that no one trusts it.

Leo: Right.

Steve: And so many problems are caused by this. I mean, so many - spoofing is causing so much problem. So the idea of having, like, authenticated inbox, that's a big move forward.

Leo: Huge, yeah.

Steve: Yeah. And that can happen incrementally. I mean, not everyone has to support it. If I knew that authenticated email was being flagged by my server and routed into a different inbox, versus unauthenticated, that's valuable to me. And we can have that today.

Leo: Yeah. Steve Gibson's at GRC.com. That's where you can post a question for next week because we'll do a Q&A, and I'm sure we'll get some questions about DMARC. We'll do a Q&A episode next week. It's GRC.com/feedback for your questions; GRC.com/health if you want to read his health postings.

Steve: Oh, and there is a feedback page there, too. And I would love to have people who - either positive or negative experiences with low-carb stuff, please send me your feedback.

Leo: Very good. And of course you go there to get SpinRite, the world's best hard drive...

Steve: Yay.

Leo: ...maintenance and recovery utility, and all the 16Kb versions of this show and transcriptions of this show. All 353 episodes, they're there. His show notes. We also make audio and video available on our website, TWiT.tv. This show, we do this every Wednesday, 11:00 a.m. Pacific, 2:00 p.m. Eastern time, 1800 UTC on TWiT.tv. Watch live. We'd love it if you watch live. But if you can't, don't worry. That's where there's always a recording available in audio or video, depending on your choice. And next week a Q&A.

Steve: Yup.

Leo: Look forward to it. Thanks, Steve.

Steve: Talk to you again, my friend.

Leo: Take care.