SECURITY NOW!

Transcript of Episode #354

Listener Feedback #144

Description: Steve and Leo discuss the week's major security events and discuss questions and comments from listeners of previous episodes. They tie up loose ends, explore a wide range of topics that are too small to fill their own episode, clarify any confusion from previous installments, and present real world 'application notes' for any of the security technologies and issues we have previously discussed.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-354.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-354-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson's here. He's going to take a look at Twitter's new Do Not Track policy and answer 10 questions from you, his listeners, all next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 354, recorded May 23rd, 2012: Your questions, Steve's answers, #144.

It's time for Security Now!. Here he is, the ketogenic Steve Gibson, our Explainer in Chief and security guru from GRC.com, the author of SpinRite, the world's best hard drive and maintenance utility, and many other great freebies. And today's a Q&A; right, Steve?

Steve Gibson: Yeah, we have our 144th, so that's 12 squared for those of you who are math people, our Q&A on Episode 354. And a bunch of interesting news, great questions from our listeners, a little bit of errata, paraphernalia stuff. So overall, I think we're going to have a really nice podcast.

Leo: You know, it's people - maybe I just don't usually get the email. But more and more I'm getting tweets and email from people saying, oh, I've got something for Steve, I've got something for Steve. They really - and I see you have some of this, the LastPass Wallet and so forth. People really are kind of, I think, really starting to know about the show, which we like.

Steve: Yes. I don't know, maybe it's my participation in Twitter.

Leo: I think so.

Steve: But I get a ton, I get a ton of stuff from people who are tweeting. And I am noticing that I think my message about please follow me if you want a response - because, I mean, I read every single tweet that I get. It's sitting here, like right in front of me, right now, this huge spread, in TweetDeck, of all these columns coming in. So...

Leo: You're crazy.

Steve: It is, yeah, it's a little much. But...

Leo: I think that might have something to do with it. I really do. I think the more you engage the social networks, the more they engage back.

Steve: Yeah, and I have to say, somebody did tweet to me, "Steve, you're really missing out not being in Google+." And I'm kind of there, but I don't know what it is. I mean, I kind of looked at it, and I didn't know what it was. So...

Leo: It's more of a long-form Twitter. If you wanted to do longer posts and engage in conversations about that post, because it has threaded responses, then it's similar, same idea. I like it.

Steve: I love the fact that I'm limited, and so are they, because it's like, well, we can't have a conversation, but here's a link.

Leo: In that case, you may not like Google+ all that much. All right, Steve. I've got your notes. I'm ready to kick things off, starting with Twitter, of all things.

Steve: Yep. There was a nice explanation, which I want to share with our listeners actually sort of the meaty beginning of it, posted by someone from the Electronic Frontier Foundation, the EFF, that is our privacy watch guard nonprofit group that is doing so much good stuff, explaining exactly Twitter's recent announcement that they are going to support the Do Not Track header which is increasingly available in browsers. Microsoft's IE has it in sort of a flavor. It's not yet supported natively in Chrome, but there's an add-on that you can get for Chrome that will add the DNT header to Chrome queries. And it is and has been supported in Firefox for some time. So the EFF guy wrote, and there's some really interesting tech stuff here:

"Under a new policy announced recently, Twitter will be suggesting accounts for Twitter users to follow based on data collected from an individual's browsing habits on websites that have embedded Twitter buttons." Okay. Let's stop for a second. What that means is, this is exactly the third-party cookie issue that we've talked about often. That is to say, when you're a Twitter user, your browser will have a Twitter cookie associated with it. Then when you go to non-Twitter sites that have the embedded Twitter button in the page, when your browser displays that page from that website and displays the Twitter button, it is sending a query back to the Twitter servers containing the Twitter cookie - which in this context, since you're actually on a different site, that's technically a third-party cookie. But what that means is that Twitter is able to track you, your movements, around the 'Net. This is...

Leo: Oh, we're going to hear howls.

Steve: Yeah, I know.

Leo: That is huge.

Steve: It is. But what it then means is that, when you go back to Twitter, it knows where else you have been and is then able to recommend people for you to follow that are relevant to what they're determining about you.

Leo: Now, that's only if you use the Twitter in the browser.

Steve: Yes, I think that, yes, it would...

Leo: Because if you were using a Twitter app, especially one that Twitter didn't write, like you - well, you use TweetDeck, which is a Twitter company. But I presume there's no way they would be able to track you. Maybe they would. I don't know.

Steve: They could certainly, I mean, there are cookies associated with your account. And it might be that, like, the Twitter apps which are not browser clients are also transacting the same cookie.

Leo: It might be part of the API, who knows.

Steve: Because you're logged in, and they know who you are. And so they could certainly give that identity. So EFF, continuing, says: "While this is sure to garner scrutiny" - exactly as you said, Leo.

Leo: Howls, yeah.

Steve: Uh-huh, "...from the press and public, Twitter is also taking a pioneering step toward respecting users' privacy choices. It has committed to respecting Do Not Track, a simple browser setting users can turn on to tell websites they don't want to be tracked. Often framed as a signal from users to behavioral advertisers, Do Not Track isn't actually about ads we see online. It's about user control over tracking of our web usage that could be used to build an intimate portrait of our online lives.

"Twitter is showing an inventive way that websites other than behavioral advertisers can respect Do Not Track. We're heartened" - this is the EFF, that isn't often heartened by anything. "We're heartened to see this forward-thinking approach and hope other sites" - well, and of course we know what this is. This is also Twitter recognizing the controversial nature of this and being preemptive about offering some choice to users who are concerned about this from a tracking and privacy standpoint. "We're heartened to see this forward-thinking approach and hope other sites with embedded widgets will follow suit." Because think about it. All of these sites that have little embedded buttons that we're now seeing everywhere, they're all doing this, too, but perhaps being much less forthcoming than at this point Twitter is.

"If you haven't done so already, this is a great reminder to turn on Do Not Track. Twitter has a tutorial for doing this on different browsers. Here's how the suggested accounts will work under the new Twitter privacy policy: When you browse around the Internet to pages with embedded Twitter share buttons, Twitter is able to collect a certain amount of information about you through a unique browser cookie." And remember, it's not just that you were there; but, because it will contain the full referrer header, it's also how you got there. So they know, for example, if you were searching for something and clicked on a search link, typically what the search query was. I mean, there's a lot of information that these third parties get. And so Twitter is setting themselves up by having these embedded Twitter buttons to collect all of this when people go anywhere to third-party sites.

"Twitter is able to collect a certain amount of information about you through a unique browser cookie. When you sign up for or log onto Twitter, the site will be able to suggest that you follow the accounts of individuals who are popular among others who visited the same sites as you." So now we're getting sort of the social networking thing. It's, gee, people who also have the same sort of site-visiting profile are following these people. Twitter can, of course, see who you're following. So they're able to pull that information, not only from you, but from others, and cross-correlate it and say, hey, here's some people that are maybe relevant to you.

Leo: Now, are these paid advertise- I mean, my question is, if it's just doing a better job of suggesting, okay. But are they paid advertisements, would be my question.

Steve: No, it's just improving its suggestion technology.

Leo: Well, I don't see anything - I'm not sure if I see anything wrong with that. I mean...

Steve: No, I agree.

Leo: You can ignore the suggestions. And if it really does give you suggestions that are good, based on things you're interested in, like it just suggested Yoko Ono and Michelle Obama to me, and Jimmy Kimmel, and Nathan Fillion. I mean, yeah, I would like to be following those people. So I don't know if - is this implemented now? Or is it going to be implemented?

Steve: No, it's in place now.

Leo: Okay.

Steve: So Twitter calls this "tailoring," in quotes, your Twitter experience based on your web browsing history.

Leo: Seems good.

Steve: I know. I think it is. "For example, many of those who visit BoingBoing.net likely follow the account of @doctorow, the digital-rights-loving BoingBoing founder Cory Doctorow. If you sign up for Twitter, and you've got a browser cookie from Twitter showing that you recently" - and this is a key, too, we're about to talk about expiration - "that you recently visited BoingBoing, you might see @doctorow listed as a suggested user even before you've started interacting with Twitter." So that is to say, it can know things even before you get going.

"Twitter will be relying on data collected from your browsing habits within the last 10 days. After 10 days, they start discarding data. When you start a Twitter account, you'll have the option to turn off the tailored suggestions. Unchecking this box won't just stop the suggestions from appearing, it'll actually remove the unique cookie that Twitter uses to track your browsing habits and show you tailored user suggestions."

Leo: That's good.

Steve: And that's, oh, they've really done it right. And that's right in the UI. It's a checkbox. You can say I want this or not. "Established Twitter users may find suggested users under the 'Who To Follow' sections of Home and Discover. Just like new users, established users can uncheck the 'tailor' Twitter box in their account settings to stop the data collection about their web browsing. Do Not Track makes this a lot simpler, no messing with account settings or unchecking any boxes to keep your privacy. If you've already got Do Not Track selected in your browser settings, then Twitter assumes you just don't want them collecting details of your online browsing habits in an identifiable way. Users who have turned on Do Not Track will find, upon signing up for Twitter, that the 'tailor Twitter' button is automatically unchecked by default."

Leo: Wow, wow.

Steve: I mean, I know they've really done a nice job. "Similarly, established users who had Do Not Track already enabled in the days before the new policy took effect will also find the account personalization turned off by default. Users who enable Do Not Track must change their privacy settings manually if they want the 'tailored' Twitter experience to be reenabled."

Now, there is a textbook example of doing everything right. I mean, just across the board. And this is why, as our listeners know, I have been so bullish about this from the beginning, about the Do Not Track header. Everyone argues, yeah, but it doesn't really do anything. It doesn't block anything. It just says. It's like, yes, but this is the way we're going to get there. Because maybe it'll get enacted through legislation, or maybe it'll get enacted through this kind of social pressure. But, I mean, Twitter doing it right sets a template for how to do it. And Twitter's not small.

So they're offering a real benefit, as you identified, Leo. It's like, wow, this does provide Twitter with valuable insight. Yet it gives the users absolute control. And if we have taken the time to turn on Do Not Track, it respects that. And if we want then to have the benefit of Twitter knowing more about us so that it can make better suggestions, and we're wanting to mature our follow lists, then we go and turn that on. It's just - it's beautiful. So I hope this is an example of the future because this is the way it should be done. And it's all we need, if everybody does this.

Leo: Can you think of any negative, I mean, of leaving it on? I mean, there's I guess the presumptive privacy invasion. But are there maybe hidden consequences we're not aware of?

Steve: Well, okay. So the skeptics will say, oh, well, how do we know Twitter's not selling this information? It's like, well, you've got their privacy policy. They don't want to get sued. Privacy is an increasingly large concern. I mean, thanks to Facebook that keeps stumbling with privacy, everyone is learning people really are caring about this. So I don't think a company is going to say one thing and do another. That would just hurt them too much. And then these things do get discovered. Employees leaving or mistakes being made, it's like, wait a minute, you said you weren't collecting this. Now some hacker just got a database of 500,000 records that has this. So what gives? Very much the way Google got caught flatfooted with collecting all of the unsecured WiFi information. They really didn't intend to use it for anything, but they had it.

So my feeling is Do Not Track may become too powerful. I mean, what I'm looking for now is the equivalent of NoScript for Do Not Track, where I can say - I can, like, have my default be not to track me; but some sites, like Twitter, I may want to track me. So in the same way that I want some sites to run scripts on my browser, but I don't want them all by default to run it. So I imagine we're going to see that evolve in time. As people support Do Not Track, we're going to want to say, well, I can't keep turning it on and off all the time. I need to selectively permit tracking. But again, this is where we're headed. This is the future. So bravo to Twitter. If we've got any Twitter people listening, they just really did it right.

Meanwhile, a trojan that we never covered, and I'm not sure why we didn't, I just probably had too many things to talk about that week, is the so-called DNSChanger trojan. And it would have been a perfect topic for us, so we'll briefly cover it now. It's something which has affected hundreds of thousands, if not maybe a million computers. And it's rather clever. And we have all the understanding, thanks to the podcast's history, to get a grip on what this thing does. It's a piece of malware which can infect people's machines. And its action is to change their system's default DNS servers to malicious DNS servers. And our listeners know what that means. That means all the lookups you do are subject to bad guys messing with the IPs that your browser receives in return.

So, I mean, we have talked about, like, DNS server hijacking, where the bad guys pollute the records of a given DNS server. And that's bad because then everyone using that DNS server who looks up a specific domain gets pointed at the wrong IP for all kinds of mischief. Well, this is, like, way worse. This is your entire computer, and in some cases DNSChanger is able to go in using, unfortunately, default logins, and sometimes Universal Plug & Play that we've warned everyone to turn off, are able to get into your router and change your router's DNS, that then affects all the machines in your network. So even though only one machine gets infected, your entire network is now receiving malicious DNS replies from these bad servers.

So this allowed people to do ad hijacking and get up to all kinds of mischief with people. The bad guys were found back in November of 2011, so about, what, six or seven months ago, and they were located in Estonia. They were arrested. And the people who do BIND, the major DNS server on the Internet, ISC, were given control of this set of rogue DNS servers.

So here's the problem. All of these machines, more than half a million, right now, today, more than 500,000 are still querying these malicious DNS servers. The problem is that the judge that was involved with this originally set a March timeframe for the DNS servers to be taken down. They're still up and running, although now they're under benign control, so they're not delivering malicious results. But more than half a million people's networks and computers are using them. So if they're shut off, everybody who's still using them loses all their Internet connectivity and has no idea why.

Leo: You know, when I read about this, I meant to ask you about it. It seemed like it was a terrible idea in the first place to replace them. It was - so people were still hacked; right? But instead of pointing to a malicious DNS server, it was an FBI-run DNS server.

Steve: Well, actually, as I understand it, they were the DNS servers that were just - they were the DNS servers taken over. So they were once malicious, but they were reloaded with good records. And so in order to de-pollute them...

Leo: Well, I don't know if it was the same machine or not. It's the same IP address.

Steve: Okay. Well, yeah, exactly.

Leo: Doesn't really matter. But wouldn't it have been better just to turn them off and say, in other words, people have been hacked. And you're kind of hiding the fact that they've been hacked. And it bothered me that the FBI decided that this was the way to handle it.

Steve: I agree. It's sort of a strange, like, I mean, this is unique, as far as I know, in terms of, like, the way we've dealt with and mitigated this problem.

Leo: I thought it was appallingly stupid. But...

Steve: So what Google has figured out is that their page, their servers are able, with very high accuracy and negligible false positives, to detect when anyone visiting their site currently has the malware installed and/or is pointing at the wrong DNS servers. And they are going to, in fact, I think maybe they have now, begun notifying everyone who has got this problem who visits Google.

Leo: That's at least - at least they'll be notified. I mean, they've been sitting there for four months with - and the thing is, if you have DNSChanger, you probably have other malware on your system. And what would have been better is just to turn it off, and suddenly the system stops working, and the person goes, oh, my, something's wrong, and gets it fixed. Instead, they've been going along for four months with, I would be willing to guess, in 90 percent of the cases, malware-infested systems that seem to work all right. Makes me mad.

Steve: Right. And what happened was this March deadline was extended to July 9th, which is - so this is, where are we, we're May, June, July. So it's coming up in another six weeks. So it's been extended because too many people were still using them. Now, ISPs were supposed to take responsibility. But ISPs didn't, for whatever reason. Clearly, ISPs could have put up intercept pages or noticed when their customers were making DNS queries to these specific IPs and sent them something in the mail. I mean, there are many things that could have happened. But it just didn't happen.

And so Google has said, okay, it's multilingual because only 50 percent of the affected users are English speakers. The other 50, located all over the globe, are non-English speakers. So Google has had to do this in a multilanguage solution. Also, Google has said in their blog posting they would like to be doing helpful things like this all they can, except that they're not often enough, and not as often as they would like, able to be accurate enough in their determination of problems. We've seen Google, for example, now warning people on links that are believed to be malicious. It's like, wait, this looks like this site is hosting malicious content. Are you sure you want to go there? So you click it, you get an intercept page saying, oh, ooh, thanks for catching that. In this case, they have developed a technology that will have a sufficiently, like, near-zero false positive. And they've been testing it long enough to know that they'll be able to deliver more than half a million of these notices within one week.

Leo: Wow. Wouldn't it have been better if the FBI had just put up - changed the server in such a way that it redirected you to a single page that said, you have malware on your system.

Steve: That's exactly what I think. I don't know why any query didn't bounce you to an FBI site specifically to present you with that notice.

Leo: In effect, the FBI has perpetrated a man-in-the-middle attack on hundreds of thousands of machines for the last six months. It drives me crazy. I don't - it doesn't make any sense. And by the way, if you go to the DNSChanger Working Group, which is an FBI front, they actually...

Steve: [Laughing]

Leo: I'm a little pissed off. They give you really loose, useless information and have buttons that say "Fix It." It's just stupid. First thing you ought to do is - they give you how to clean your PC against DNSChanger. And I just - I feel like - now, actually this has gotten better than it was when they first announced this about a month ago.

It just seems suspicious to me that they didn't do that, that they decided to keep it running for a while. We'll do your DNS for you, just, eh...

Steve: Trust us.

Leo: As you were.

Steve: Speaking of "trust us," RSA's - I'm really tempted to put the syllable "in" in front of "SecurID," making it inSecureID. Windows software tokens are deep in the doghouse. Now, we've talked about the hardware tokens extensively, the little dongles. We were pioneering the PayPal football and eBay logon, the famous football for providing third-party security. I, of course, love what YubiKey has done with their little one-time USB token. So RSA couldn't resist doing this in software.

Well, it turns out that we have another classic example of good crypto implemented poorly. Not the crypto's fault. The architectures are absolutely solid as could be. But it turns out that their Windows - they have a Windows version of their SecurID token so that people who don't want the dongle can install this software. Now, already we know, whoops, bad idea. The whole point of the dongle is, well, one is portability, so you have cross-machine portability. But the other is that it's not connected to anything. I mean, it's not in your system. The YubiKey is safe because it just emulates a keyboard. It will not receive any commands to allow it to dump its information. Even using the YubiKey UI technology you can't get that information. It will not give it up. So those guys did it right.

RSA said, oh, well, let's sell people a software token. Think of all the money we'll make. Well, it turns out that a security researcher discovered that the serial number, the required token's secret serial number is simply determined by a combination of the machine's host name and the current user's Windows security identifier, which is stored in the computer. So, busted, completely. It means that, if any software is able to get into your machine, and you have one of these, it can instantly clone your SecurID and, at will, generate exactly the same sequence of supposedly cryptographically secure, but cryptographically generated now, but no longer secure, six-digit numbers that the SecurID software does. So it's completely reverse-engineered, and we hoped it would be a secret serial number, turns out not to be. It's derivable from information sitting right there in the machine. So, whoops. And there's only 40 million people using this.

Leo: Oh, geez. So this is - is this in Windows? Is this - who's using this?

Steve: This is in Windows.

Leo: How would you know if you're using it?

Steve: Oh, you would be a user who periodically calls this up. And it comes up in a little - it's got its own little Windows UI with a little screen that shows you your current SecurID six-digit token.

Leo: Ah, okay. So it's like that Google - the Google Authenticator or the football that VeriSign provides.

Steve: Right.

Leo: Same idea.

Steve: Same exact, I mean, it's the same algorithm, same everything.

Leo: Just poorly implemented.

Steve: Well, written in software. And, for example, they didn't use the Trusted Platform Module. The TPM that's in most laptops is sitting there for this purpose. But someone said, oh, well, most people don't have that turned on.

Leo: Right, or don't have it.

Steve: Or most people aren't using it, or don't have it. So we won't use it. But what they could have done is arbitrarily created a non-derived pseudorandom number through some handshake. I mean, again, it's difficult to see how you can protect this. This is fundamentally a bad idea.

Leo: Well, that's what makes me wonder, I mean, Google Authenticator software on my mobile device. So I wonder if it has the same flaw.

Steve: Well, and I've got a VeriSign VIP token on my BlackBerry which is able to generate numbers for me to log in in Windows. But those are separate platforms. So, I mean, it's certainly the case that doing this in software on a shared software environment will not be as secure as any little piece of hardware that is just all it's going to do is give you numbers. So, but the tradeoff is convenience. But again, putting it in Windows and then using...

Leo: For logging into Windows. Right, right.

Steve: Or your corporate VPN. I mean, you just - so the point is, this is a huge vulnerability for anyone who depends upon it for security. It simply never was secure. And now that the world knows how to hack it, and all the details have been posted, it's just blown. It's just completely blown. If your machine ever got infected, your Windows machine, the malware could grab that data trivially, send it off to wherever, and they now have the ability to generate the same secret token, the same token stream as you do. And there went the security and the authentication, everything it purports to provide. Dumb. I mean, it's just a bad idea to offer that on a Windows platform.

Leo: Yeah.

Steve: Now, in happy news, LastPass has released a free secure Wallet for iOS devices.

Leo: I saw that.

Steve: Yeah, and it looks very nice. It's built around the existing LastPass feature called Secure Notes which LastPass has always had. LastPass for our browsers can store more than just userID and login. You can have it store notes for you also. In the case of Wallet, which is 100 percent free forever, it comes with a series of templates for credit cards, passports, driver's license, memberships, bank accounts, and a bunch of more things. Current LastPass users, as so many of our listeners are, can login with their usual account information. So this is sort of like a - it's a client for your existing LastPass account and entire database. And any existing Secure Notes will automatically be synched to the Wallet, where they could be viewed, edited, and synched back through the cloud to all other locations.

Leo: What does it mean when a baby toy goes off?

Steve: That's my phone. I forgot to silence it.

Leo: Your telephone is a squeak toy?

Steve: Yeah, it's a squeaky toy.

Leo: Okay.

Steve: That's generic, unfiltered incoming email. It just makes me laugh. And anyone who I'm, like, at lunch with or something will laugh.

Leo: It is funny. I'm laughing.

Steve: Yeah. So audio clips, photos, and texts can be saved as an attachment to any note. You can use the app itself to record voice clips and securely store pictures from your device's photo gallery. And in a blog, in the announcement blog, our old friend Joe Siegrist, who gave me all the information back in the day when I was able to fully vet LastPass and adopt it for myself and recommend it without hesitation, he responded to somebody who was grumbling about why they didn't do it for Android first.

And he responded saying, "The reason Apple was first here is that Apple does not allow us to do a free trial. So it appeared that the best way around that problem was simply to give away something free" - instead of, like, a trial period - "to give away something free that gives people a taste of what the full LastPass does on their Apple device. When you tell your friends at a party that they should be using LastPass, Wallet is a good place to get started if you're an iOS user. We really had nothing for that case before, for the Apple platform."

So that was Joe, the main technologist guy there, explaining why. And which makes a lot of sense to me. I just a minute ago grabbed it. I haven't begun to play with it yet. But I've got it now added to my various iOS devices. And I trust these guys. I know they know how to do it right and the importance of getting it right. And so I wanted to let everyone know that this looks like just a pure and simple win, especially if you're already a LastPass user, to have an iPhone interface or an iPad interface.

And in the Anonymous Mailbag, I found something coincidental that I thought I would share. From a Richard S. in Burbank, the subject of what he sent me was "Elaine's transcripts extremely useful." And he said, "Dear Steve, just an anecdote demonstrating how useful it is to have the transcripts that you have asked Elaine to produce. I was sitting at a research retreat with a very computer-savvy MD who was typing his secure password into our institutional VPN. I asked him whether he knew about LastPass. He said no. I explained it to him and showed him my one-click login to our VPN. He was impressed and asked if it was secure, and I said yes, absolutely. The computer-savvy MD sitting next to him said that LastPass had been breached. I said no, but MD No. 1 starting Googling the issue and getting alarming hits. So I Googled "LastPass breach Steve Gibson GRC," and immediately hit the transcript for Episode 301, which I forwarded to him. He read it and downloaded LastPass. An invaluable resource."

Leo: Awesome. Well, I'm so grateful that you pay for that, and that Elaine does it. It's really, it is, it's a huge - if we had the money, we'd do it for every one of our shows. It's a huge value. Thank you.

Steve: Well, and not only is it, as we've just said, well, the primary motivation was that people like to read along, or just sometimes read and not listen. So this gives them an option. But the other obvious thing that's happening is it makes our otherwise audio podcasts or video podcasts searchable. You can find them. And so you could, yes, you could then download the transcript, or it finds the audio and/or video for you also. So very cool.

Leo: Excellent point.

Steve: An anonymous listener also said, "Steve, don't switch to Mac just for ARQ." This is my Notes from the Cloud section. I saw that, and that piqued my interest. That was the subject line in the Mailbag, so I dug in. Because it's funny, I tweeted that ARQ for the Mac was so good as a really nice frontend for Amazon's S3 that it made me want to switch.

And so he said, "It was great to watch you zero in on the right architecture for cloud backup: A, fault-tolerant pay-as-per-usage storage backend like S3; B, a trustworthy frontend with strong crypto implementation like ARQ. But there is still hope for PC users, and Linux and Macs, too. Check out www.duplicati.com. Everything you talked about ARQ, and cross-platform, open-source, non-Java, native code, plus plug-ins to many multiple backends. Would love to hear your analysis on any crypto and other weaknesses possible in Duplicati code and architecture."

Leo: I love it that it's open source.

Steve: I do, too. And so under Features they list: "Duplicati uses AES-256 encryption or GNU Privacy Guard to secure all data before it is uploaded." So it's pre-egress encryption, pre-Internet encryption. "Duplicati uploads a full backup initially, then stores smaller, incremental updates afterwards to save bandwidth and storage space." And on their pages they go into some detail about how this is done.

"A scheduler keeps backups up-to-date automatically. Encrypted backup files are transferred to targets like FTP servers, Cloud Files, WebDAV, SSH through SFTP, Amazon S3, and others. Duplicati allows backups of folders, document types, e.g., documents or images, or custom filter rules. It's available as an application with an easy-to-use user interface and as a command-line tool." Oh, 100 percent free, by the way, if I didn't mention that already.

Leo: Well, you'll pay for the storage.

Steve: Yes, but not for the frontend. "Duplicati can make proper backups of opened or locked files using the Volume Snapshot Service (VSS) under Windows or the Logical Volume Manager (LVM) under Linux.

Leo: Interesting.

Steve: And then it gets better. From their How-To: "Duplicati is built using standard tools and formats. Un-encrypted archives are simple .zip archives. Encrypted archives are .zip archives that can be decrypted with AES Crypt. So even without Duplicati," they said, "all your data belong to you." And it supports 1&1 SmartDrive, Amazon S3, BACKUP.ACtion, Box.net, CloudSafe, DriveOnWeb, Google Drive, SkyDrive, Strato HiDrive, Tahoe-LAFS, whatever that is, and T-Online Mediencenter.

So I'm very impressed. Their future plans, they're at v1.blah blah blah. Their 2.0 will add running as a service so that the whole thing can run when you're not logged in, doesn't need to run as a user-launched app, and they're also going to reengineer the UI. So everything about this thing looks right. And so I wanted to let all of our listeners know that, thanks to an anonymous mailbox person, we now have knowledge of yet one more interesting multi-backend frontend.

Leo: There's just an infinite number of these, it seems, as time goes by.

Steve: Yeah, well, it's just - yeah. And here again, this is - this fits my model. My feeling is this is not something that people should charge for, that is, the privilege of having a frontend to Amazon S3. It ought to be free. There's just not enough value added to make it something that you should pay for.

And I wanted to mention, in a GRC R&D update, I hit a milestone yesterday with my main project that I've been working on and have referred to a little bit through the year. I started late last year and just fell down, I don't know what kind of hole it was. And actually it turned into an amazing R&D adventure. And as I had mentioned, I have solved a fundamental problem in computer science, which has never been solved before. As far as I know, I mean, it is an invention. It is completely unique. It is dramatically more efficient than any other approach for solving this problem.

And I will be - I'm not going to patent it or claim any IP. I'm going to get it documented, and we'll do a podcast on it. And this is my solution for finding in really, really large corpuses, which is to say big blobs of text, the longest strings that repeat, anywhere in them, in these corpuses, without knowing what they are. And there are well-established solutions, but they are only toy solutions. You can only do them on small little data sets because they just completely collapse and break. They do not scale. And I've come up with a brand new solution that no one ever seems to have seen before - I haven't, and none of the other smart guys who have been following this have - for, like, really solving this problem.

Then I was trying to figure out how I could explain this because there's a lot to it. And what I realized was I needed - I couldn't do it in one podcast. I needed to establish some additional foundation that we've never had before, which is sort of back to our basics. Our long-time listeners will remember that we did a series of how computers work, sort of from first principles up. Starting at the beginning, we went through the entire evolution of the way computers function. And I stopped at the instruction set as, like, okay, now we're there, like different types of instruction set. We talked about CISC and RISC and so forth.

But the next layer above that, which is below solutions but above instructions, are data structures, that is, pointers, arrays, records, stacks, trees, and lists. That is, these are the things that computer scientists have come up with which are implemented sort of universally on any computer, using any instruction set, to sort of create the first layer of abstraction above the instructions. Yet they are universally applicable to all kinds of problems. So, like, anyone who's gone through computer science has learned about pointers, arrays, records, stacks, trees, and lists. These are - and you know all those words, Leo. These are fundamental ways, like fundamental tools which you use for solving, for them solving problems using those.

And so we will first do a series of podcasts, not a long one, maybe two podcasts, I think, ought to be enough to cover those because I have used those, and I realize I can't bring all of that information along, and what my LRS (Longest Repeating String) technology does, at once. So we've got a bunch of fun podcasts here in our near future.

Leo: I see that it's a problem people want to solve. But why? What would you do with the longest repeating string, substring?

Steve: Well, I got...

Leo: Why would you search for that?

Steve: I got into this because I've got completely disorganized SpinRite testimonials. And when I share them with our listeners every week, I'll sometimes - I'll move them, like, into a big list I've been collecting. But there's a bunch of them already on the website. There's a ton that are still in email. There's a bunch I've never found that are in the Security Now! mailbag. And my concern is I want to pull them all together and then eliminate duplication. I don't want to have any that in there twice. And so, but sometimes, for example, before I read them, I'll fix some grammar, or I'll change the spelling of SpinRite if it was W-r-i-t-e, for example. So I'm not going to get exact matching. I needed soft matching of, like, phrases.

Leo: And if you did shortest repeating string, you'd get a lot of periods, spaces, A, B, C, Ds, and Es. So you want longest because that's the biggest match.

Steve: Correct, exactly, the longest matches. But then other people who have had insight said, wait a minute, Steve. You can use this for DNA matching.

Leo: Oh, for lots of things, yeah.

Steve: And, it turns out, for instantly decrypting texts in unknown dead languages. If you dump a text into this, it'll instantly find all the words and all the phrases using repeating series of words. I mean, it just organizes disorganized stuff instantly. Well, not instantly. But, like, in a practical speed. And nothing like this exists as far as...

Leo: Even in grep and things like that? They don't do that, huh?

Steve: No, because you have to know what you're looking for. Here you don't have to know. It finds the patterns for you.

Leo: I cannot wait.

Steve: It's very cool. So speaking of SpinRite testimonials, I had a really fun story from a Christian Alexandrov in Sofia City, Bulgaria. And he said, "Hi, Steve. I want to share a SpinRite story with Security Now! listeners. My cousin was traveling on her way home in her car when she hit a pothole" - apparently this was big pothole, maybe they have big potholes in Bulgaria - "and broke the suspension on her right front wheel. The car lost control and spun off the road. The good news first: No human casualties. Bad news second: Her laptop took a pretty good hit. The plastic body was full of cracks. She came to me and said, 'I gave the laptop on a few PC repair services, but all said it is already dead. I'm desperate. See if you can bring it back from the dead.'"

He says, "I was looking at this broken laptop and gave up any hope on it. Anyway, I tried to boot on battery. No boot. The battery was broken badly. I disposed of it. I tried to boot on its charger. I got luck, but not much. It gave me blank black screen and, a few minutes later, failed attempt to find OS. So I used SpinRite on Level 2, but no luck. I was about to tell her this corpse is dead beyond resurrection.

"Then it hit me like lightning. I ran SpinRite on Level 1 to get the drive to know itself again and force it to take action on itself using the tip you had recently mentioned and suggested on Security Now!. After Level 1 completed, I tried to boot, and at least now I was not waiting for a few minutes to try to boot. It started to boot normally, and I saw the BSOD saying 'Unmountable boot volume.'

"I got upset, and I decided to go full force. I ran SpinRite at Level 4. It took 96 hours straight to finish the drive, reporting a lot of green 'R' icons and few red 'U' icons. So many sectors recovered, and a few at least partially recovered. I ran SpinRite again on Level 4, and now it only took a few hours to finish the entire drive. I saw a lot of 'B' icons" - meaning that SpinRite had marked those sectors bad and relocated the data - "on all places where the 'R' and 'U' icons were before. I grabbed the chance and quickly backed up the entire movies folder. This was the only folder my cousin desperately needed saved. So I saved this folder to my PC. When I checked its contents, I was stunned when I saw what was in there: the complete first three seasons of 'Farscape' on high-quality."

Leo: Thank goodness we didn't lose it.

Steve: No, we didn't lose those grasshoppers or whatever they were on "Farscape." They were strange creatures. "Needless to say, I kept a copy for me to enjoy a fine sci-fi series. I disposed of the hard drive of this laptop, bought and installed a new one, reinstalled Windows, and copied back the movie folder. My cousin was as happy as she could be. Thanks to you, Steve, I have the chance to enjoy a fine sci-fi series. If we get to meet face to face, I promise I will buy you a bottle of red wine on my expense, the finest Cabernet I can find on Bulgarian market, regardless of the price. Thank you, Steve, for this great piece of software, and thank you Steve and Leo for the great Security Now! podcast. I wish best of luck to you, GRC.com, and TWiT.tv from a happy SpinRite user."

Leo: Well, I just hope he turned the BitTorrent client back on when he restarted the machine, that's all.

Steve: So thanks for sharing that, Christian.

Leo: That's pretty funny. To the Q&A we go. Starting with our first question from Michael Dombrowski. He is a smart kid, a high school sophomore in Washington, D.C. And we were talking about ARM versus x86. And then he wants to know about Windows 7 and 8. First of all, love the show and watch every week, as well as the past two weeks of the "Sugar Hill" series. I wish I could get my high school student to watch that. He loves his French fries. And I hate - every time I see it, it, like, pains me. I see him eating them, that pains me.

He says: I've been thinking this through in my head for a while and wanted you to tell me if I'm right or wrong. Or crazy. I was thinking that because the Intel x86 architecture and ARM both have the same fundamentals - AND, OR, and NOR gates - would it not be possible to map all of the x86 functions into the reduced set that ARM offers? In other words, get to CISC from RISC? I remember you saying in the fundamentals of computing that x86 just kept building in calls for things like multiplication, for example, and that ARM had a limited number of these. But because they both must run on the same kind of logic gates, then could someone not implement in hardware or software an X86-to-ARM translator?

From what I know, which is limited as far as hardware is concerned, this would be possible. It may take a lot of work, but for someone like Microsoft, maybe no big deal. Sorry for the length of this email. I wanted you to understand my question and make it as clear as possible. Thank you for all that you do. I feel I'm getting a comp sci class even before I go to college. Michael.

Steve: So that's very clever and has absolutely been done historically. There are two things, two approaches that could be taken. For example, you could literally perform a static translation, where you take a program written in one instruction set like x86 and, instruction by instruction, convert each CISC (Complex Instruction Set) instruction into the equivalent series of RISC instructions, and essentially do a static translation from one instruction set language into another. So that can be done.

But what has ended up being done is a variation on that, and that is an emulator. And emulations exist all over the place because it's ended up being a powerful thing to do. In fact, historically, you'll remember, Leo, back in the days of early Pascal with p-code, the Pascal compiler generated a pseudo-code, which is what the "p" stood for, p-s-e-u-d-o, pseudo-code, for an imaginary computer that didn't even exist. And then somebody who wanted to run Pascal on a given host platform, on a given computer, they would write an interpreter in the native machine language of that processor, which interpreted the pseudo instructions that the Pascal compiler produced.

And the beauty was that you had a huge library of Pascal programs. And in fact the actual environment for editing and compiling Pascal was written in itself, in Pascal. So when you simply wrote this layer, this interpret to interpret p-code, as it was called, suddenly everything works. You get all the programs, all the software, and a little operating system that's ready to go and does go.

So it's certainly the case, for example, that Windows could have been left in x86 architecture, but Microsoft could have implemented an x86 emulator which would run on the ARM architecture and emulate the x86 instructions one by one. The problem with doing that is performance because you're not directly executing the native instructions. There is that layer, that emulation or interpretation layer, and there's a big performance hit. I mean, it varies depending upon how closely the architectures map into each other. If the ARM, for example, had enough registers to simulate the x86 registers, then that would help. But if the architecture didn't, then you'd have to simulate registers in memory. So suddenly, you can see, that would be a lot slower than emulating registers that were in the actual hardware architecture.

So to the degree that there's mismatch between what the architectures do, that creates more friction, sort of, between them. And since Windows is almost 100 percent, if not now 100 percent, written in high-level language - C, C++, or other abstractions of C and C++ - it's very easy, comparatively, just to recompile the existing high-level language implementation of Windows. Rather than using an x86 compiler, you use the now very mature ARM compilers. And you end up with Windows running directly, natively, on the ARM architecture.

So, Mike, yes, it absolutely can be done, has been done. There's a history of it. Java, that we talk about often, is a so-called "virtual machine," the Java Virtual Machine, the JVM. The Java compiler produces, again, this so-called "bytecode," or a specific sort of virtual language. And then you have different virtual machines, JVMs, one for the x86 on Windows, one for the x86 on Mac, one of the old PowerPC on Mac, and so forth. And then when you implement that JVM, the virtual machine, all Java-ness runs on top of it. So it makes porting and portability very nice at a cost of performance that you can't get around. But sometimes that's a tradeoff that really does make sense.

Leo: So you could cross-compile, which would be kind of a one-time-only translation.

Steve: You really could.

Leo: Or what you're doing, which is kind of an on-the-fly, as-you-go, real-time translation. The interpretation is a little slower, but it's a little more flexible, obviously.

Steve: Well, and I guess the way, yeah, the way to think about this is, if you were to cross-compile, then you're taking the...

Leo: That's like a one-time translation.

Steve: Right. But the way to think of it in terms of the result is you're taking Windows, that was written in C, that was compiled through a compiler to x86, and then compiling it again into a compiler that compiles x86 into ARM.

Leo: Right.

Steve: So if you did not have access, for example, to the original Windows source code, then that would be really your only alternative. But since Microsoft does have access to Windows source code, and there is a C compiler for the ARM, they don't have to go through, like, a two-stage inefficient compiling process. They just go one stage. They just recompile Windows in C directly to the ARM architecture.

Leo: And then you have, as in all code, you have some tweaked stuff that's written in assembler, and you would just translate that by hand or...

Steve: Right. In Windows it's the HAL, the so-called Hardware Abstraction Layer, is the layer which is intimately familiar with what Windows - with the platform that it runs on. It knows about the way PCs deal with PCI and USB and timers and interrupts and all that. That's like the real nitty-gritty interaction with the hardware. And so you'd have to rewrite the HAL in ARM assembler. But that's a small portion of what has become the behemoth that we know as Windows.

Leo: And you'd probably do that by hand because it's a small amount of code.

Steve: You'd have good guys who do it, yeah. You'd put your best talent there because everything runs through there and depends on it.

Leo: Right, right. Chris Waters, somewhere in the U.S., wonders: How secure are aging Linksys wireless routers? Greetings. I don't believe you've addressed the following on your fine podcast. If so, I apologize. I have a Linksys WRT54G - a classic, ladies and gentlemen - v8.0. It has the latest firmware, but it's from October 2009. It's at least five years old or more.

Is there anything inherently insecure with continuing to use it? The router seems to work well, locks up every few weeks, necessitating a reboot, cycling the power. I preemptively reboot it every week. Googling this problem indicates it's quite common. Any ideas as to what is causing it and how to prevent it? I've considered installing alternate firmware, DD-WRT - I think this is the classic router for DD-WRT - or Tomato. I'm unsure how secure these alternatives are. What do you think?

Steve: So my sense is that our wireless router technology is enough of a moving target that it really does make sense to stay current. We've been talking about and will be talking about buffer bloat in the past and in the future. Here you've got firmware that's, as you said, Leo, four or five years old. And we're finding little edge case problems with, for example, the wireless easy config technology, which it would be nice to have Cisco update for their Linksys firmware.

But this is probably an abandoned piece of hardware, whereas it by no means has been abandoned by the DD-WRT people or the Tomato folks. And Tomato is a beautiful, thriving, living community and state-of-the-art piece of software. And when solutions to the buffer bloat problem occur, they will be implemented immediately in these third-party firmwares for those routers. So I'd vote for making the jump. I'd look into which one makes the most sense for you. I don't really have an opinion between those, not being...

Leo: I like Tomato.

Steve: And that's what I thought. I thought you did, and that's what I was going to say. And it's, I mean, that's everything good about Tomato. And it will be kept current. And if things evolve in the future which require change or updates - like the buffer bloat problem and the solution that we now have in test, essentially, and we'll be doing a podcast on that shortly - it'll be available in Tomato very quickly. So I'd say make the move, Chris.

Leo: And that's a classic router. That's actually the router, if my memory serves, that they recommend for both DD-WRT and Tomato. It's actually valuable because it's so easily hacked. And it does have one advantage, Steve, that newer routers may not have - no WPS. It doesn't have any of those kind of fancy built-in things.

Steve: Nice.

Leo: And you're right, that might be an advantage. Now, at some point I do want to talk about the new 802.11ac with you. And if you want to do a show on that I would welcome it. The new standard is out. Buffalo and Netgear have both shipped "ac" routers. I actually have the Buffalo router. And it has some real advantages. It's much faster. It's very interesting. Unfortunately, the Buffalo router also has WPS on it.

Steve: Hmm.

Leo: [Frustrated buzzer sounds] But at some point, if you'd like to talk about it, I'd love to hear it.

Steve: Cool.

Leo: David, and I'm not sure how - it looks like his name is mangled a little bit here with the Unicode. David Garcia-Abad in Basque Country of Spain wonders about encryption for the ultra-paranoid: Steve, I've been a listener for about two years. And although I will not go into detail about how fun it is for me to listen to - oh, please, David, go right ahead - I have to admit two things: Leo and you have made my commute time much more enjoyable; and, if Leo ever decided to charge money for the podcast, I would pay for it. Hmm. I'm that hooked. I just wanted to know your opinion about a very simple idea.

Say you have some very confidential data, and that you want to make as sure as possible that no one will ever be able to crack it, not even the Utah-based super-cracking mega-facility. Oh, he does listen. How would you proceed? Being ultra-paranoid, this is the approach I would take, nothing fancy: First, encrypt the plaintext with AES-256 using key one. Then encrypt the resulting ciphertext with a 448-bit Blowfish, using key two. Then encrypt the resulting ciphertext with a 256-bit Serpent cipher, using key three, and et cetera, et cetera, et cetera. You get the idea. Is that the end of it? I can't tell. Let me track this down.

It's kind of a Matryoshka doll approach to encryption, using different algorithms and different keys. If one of the keys gets compromised or cracked, well, we still have the others. If one of the algorithms is found to be weak over time, well, we still have the others. I know this is not the most original or convenient thing in the world. I just want to know your opinion about this possibility. Thank you for the show.

Steve: So, okay. This sort of is interesting because, from a strict crypto standpoint, it raises the question. And one of the options in TrueCrypt, the whole drive and file encryption technology that we like so much, it's got options for serializing the encryption algorithms using more than just one, just for this purpose, for the concern that, not so much keys leaking, but that's certainly a valid point, too, but that at some point in the future there might be some vulnerability found in any of, like, in the one crypto algorithm you were using, the one cipher. And that would then weaken your position. But if you used two or three or four, then no one weakness would cause a problem.

So part of David's question is does daisy-chaining ciphers in series using completely different and unrelated keys, because that's certainly what you would have to do, too, does that scale your security? And the answer is yes, absolutely, 100 percent, 200 percent, 300 percent, whatever. I mean, it is absolutely the case that you end up with the multiplicative effect, really, not even additive, multiplicative, of one cipher, followed by another, followed by another, each using unrelated-to-each-other keys. You essentially just keep summing up the key length, and you end up with a ridiculously long key, and you have a cipher strength equal to the total bit length of all these unrelated keys, implemented with taking chunks of that in different, completely unrelated ciphers, driven by that portion of the mega key, and then feeding it into the next. So you've ended up with something insanely powerful, but at a performance cost.

And it's why, for example, when I set up TrueCrypt, I simply use AES-256. TrueCrypt now uses Intel's instruction set support, so it's even faster than it was before. And that instruction set support only works for AES-256, so that gets accelerated. The other things don't. And I've looked at and torn apart and we did a podcast on AES-256. And, I mean, everything to me looks like it was nailed. So while, yes, you can always scale your crypto beyond reason, to no end, I mean, without limit. I guess I ask, why? Once you have enough, then more is just more cumbersome and slower. It's not clear that, if it's unbreakable, it can be more unbreakable. If you've got infinity, then you can't do...

Leo: Can't do plus.

Steve: ...infinity plus one. Plus one, yes.

Leo: Sam Cornn in Machesney Park poses two practical questions about the DMARC email security we talked about yesterday. One, what, if anything, will Google Apps customers need to do to enable DMARC? And, two, what would need to be done for PHP mail scripts?

Steve: Okay. So we didn't talk about that, which is why I liked Sam's question. Customers would need to do nothing because this is inherently a server-to-server technology, where servers which are accepting email from other servers will authenticate that sending server's identity in order to believe the email that they're receiving. So customers at each end, that is to say clients, the clients of those servers, just - their clients don't need to change because they've established independently their authentication with their own home server. You log into Gmail using your strong credentials to say I am me, or you log into your corporate email server with your email client, saying I am me. Very often, if it's a corporate email server, it's inside the company's firewall. And so no one else has access to it as a client except people there inside the firewall. It's not like if you're using a POP protocol on port 110, that's not available from the outside. So no non-local clients can even access it from a client standpoint. And then that server connects to the destination email server in order to authenticate the transaction and send the mail.

Now, as for PHP mail scripts, mail scripts could operate two ways. They could operate as a client, or they could operate as an SMTP server, receiving connections on port 25, which is SMTP's default port, or sending, connecting to other servers on port 25, in which case that PHP mail script is being a store-and-forward architecture. It's not being a POP client where it's connecting to another hosting SMTP server as a client. In that case, it itself would need to implement the DKIM and SPF architecture protocols that we talked about last week. So either they exist now, or they will.

For as popular a language as PHP, if there is a need for it, all the crypto backbone is already in place for PHP. You can get crypto libraries. And so either a talented PHP programmer could implement those protocols using the libraries, and they're really not that difficult to do, or wait a while, and somebody else will do it, and you'll just be able to import the library and be a DMARC-compatible server, if that's really what you want to do in your own language. So users get to ride for free. And if programmers want to tackle it, the docs are solid and available.

Leo: Cool. Question 5, Dominic Black in England talking about those older security protocols, SPF and DKIM, does not - he says they don't prevent domain spoofing, but they help: You said in last week's Security Now! that a folder saying "Authenticated Mail" would be great as you could trust it. But I counter that, saying you can't anyway. For instance, with PayPal.com, what is there stopping me from, say, buying paypai.com and then capitalizing the "I"? It'd look like PayPal.com in most browsers and email clients. I would set up all the SPF/DKIM settings on my paypai dot account, and my fake email would now appear in your authenticated mail folder. It would point you at the fake site, which I would design to look exactly like the real one, apart from an EV certificate. But then again, someone in theory could get an EV for a fake domain.

Because of your authenticated folder, now you are more likely to trust my email than you would have been. It's authenticated, but authenticated to a spoofed address, I guess. Further to this, could we simply not use high ASCII characters and Unicode characters, now that the domain name system supports it, and buy domains with characters which look incredibly close? What can you do to prevent that, huh?

Steve: Yeah. He's right. I mean, this is not something - this is sort of the human factor side of this problem. And I don't see a solution for it. I know, for example, that PayPal has themselves proactively, I mean, just to use Dominic's example, proactively grabbed similar domains to prevent this problem to what degree they can. But this is just Dominic's example. It could be any domain where you're able to create something that looks deceptively similar. And, I mean, and his logic is thorough. I mean, this is the kind of logic we develop on this podcast over time, is that by creating authentication and then abusing the authentication, we end up making a claim which, if you misunderstand the claim, appears to be stronger than if you hadn't made it in the first place.

So it's a very good point. It's not a problem that I can see any means of solving. At some point the user is responsible for looking at what's going on and being careful. But I did, I felt I had to bring this up. It was mentioned some weeks ago by a good friend of mine, and Dominic is exactly right.

Leo: It's merely authenticating the phony domain.

Steve: Right.

Leo: Ghislain in Espeluche, France wonders, if SKIM and SPF, why all the spam? I think he means DKIM. On the DMARC podcast you said that 80 percent of the mail volume is currently using either SPF or DKIM. But everyone agrees that 90 percent of the mail volume is spam. So doesn't this mean that almost all of the spam is already SPF and DKIM compliant? Therefore, this kind of authentication doesn't fix anything, and we're really making all this bother for no result at all. Am I missing something in reading those stats? Also, I can say that DKIM is a real pain to configure right. Regards, Ghislain.

Steve: So I think I was probably not clear. It's 80 percent of the valid email volume is currently either SPF or DKIM. And the problem is that, even so, we're not trusting it yet. This is the whole point of DMARC. And Ghislain mentions that it's a real pain to configure right. He is correct. That's why I haven't done it yet. I don't have a DKIM-compliant email server. My next one will be, as I mentioned. But I do have SPF configured. However, I don't know if people are blocking spoofed email from GRC.com because there's no way for me to get reports. And I don't know when I do configure DKIM, unless it were for DMARC support, I want that feedback that it is, you know, don't block this, but send me reports.

So it's really clear to me that DMARC is a step forward. But it won't be until we have enough confidence in the system that we are then willing to block email that doesn't meet the requirements. We know, for example, that PayPal, as I mentioned last week, has relationships with Yahoo! and Gmail where they said we want you to block anything. We're going to take responsibility for authenticating anything, everything that comes from us, period, for the benefit of ruling out any similar spam. But the rest of the industry has been, like, well, you know, we're not really sure about this. It seems like a good thing, but how do we test it?

And DMARC brings testing and then will, I'm sure with time, bring, I mean, it will just - the email world will switch to a mode where everything is blocked, and things will be better. Not perfect, again, as our prior questioner mentioned, but better. And we don't want to let better be the enemy of perfect. So being better is worthwhile, too.

Leo: Question 7 from Bob Carneim in Oak Ridge, Tennessee, who says: I'm going to belabor the point. I've been wanting to bring this up for a while now, so I hope this may come to your attention. When I heard Charles Hill's question in the last Q&A, I thought, okay, I'll get my answer. Alas, just as Charles thought you missed the point that Gleb Budman - Gleb Budman. Is that really his name, Gleb Budman?

Steve: I know, it is.

Leo: Try to say that three times fast.

Steve: It's not easy.

Leo: Just as you, Steve, missed the point that Gleb Budman, the Backblaze CEO was making, I think you're missing the point Charles was making, which is this: Your Jungle Disk/S3 solution, which I also use, is not integrated, and there is no inherent TNO encryption in the S3 bucket. So you are counting on the TNO encryption the Jungle Disk client application is assumed to provide. This assumed TNO is provided by the optional encryption key you type into the Jungle Disk client settings. But how do you know the client is not transmitting that key to the Jungle Disk HQ or Rackspace or Amazon or the National SA, Spy Administration?

I've been using this solution ever since you described it way back when, but it always bothered me. I would think that the only way you could be sure your key isn't being transmitted off your box is to have a separate machine inline between your box and the Internet that is watching all the packets for your encryption key to fly by. But even then, if they encrypt your key, the analyzing box wouldn't be able to recognize it. So did you or could you disassemble and reverse-engineer the client app and analyze it to positively determine that it is not sending out your key in any form or fashion? Thank you. Bob.

Steve: So this, Leo, is why you love open source, because it makes the claims potentially provable.

Leo: Uh-huh. I agree.

Steve: Yeah.

Leo: Crypto always should be open source. Otherwise you don't know.

Steve: The TrueCrypt application that we love is open source. People can look at it and have at it. Now, I would argue that it's a function of tradeoffs. Now, TrueCrypt was designed by a bunch of nice people who went to a huge amount of work to create a great solution. And we thank them for it. Maybe we donate money. I've supported them several times because I want it to keep going and stay alive and be current. And so maybe that's the model that allows this to work.

But if I were going to do a crypto solution and invest a substantial length of my time in it, it would be commercial, and it would be closed source. I'd be happy to document the protocol and the technology and do everything I can to demonstrate the way I have designed it to work. But it can't be open source if it's also going to be commercial because I want to sell these things. So does that mean I don't do a really valuable crypto product? No. It means...

Leo: No, but people trust you.

Steve: Exactly. And so Bob is absolutely right. I did not reverse-engineer Jungle Disk. But I did speak to Jungle Dave. I think we had him on the podcast, in fact, years ago.

Leo: I think we did, yeah [Episode 123].

Steve: And I was completely convinced, enough that it's what I trust and use, to use his solution. Now, we just talked about a different open source, that Duplicati.

Leo: Right.

Steve: Which is open source, a frontend for S3, very cool-looking solution. I don't know if it's going to move me from Jungle Disk. I'll take a look at it. But so we are seeing alternatives. My feeling is, if it's open source, then you're still trusting that somebody else, presumably other than you, has looked at it and not found anything wrong with it. But I guarantee you, if somebody wanted to bury something in there that was misbehaving, they could do such that your visual scrutiny of it couldn't locate it. And also you're ultimately, knowing that it's open source, but you're using a precompiled EXE.

So the only way to really satisfy yourself is really to write it from scratch yourself, which is probably not practical, or really go over it yourself with a fine-tooth comb, and then you compile that source into an EXE which you use. Anything short of that, then you're trusting people. I mean, my point is there's always going to be some trust. And of course we've got the operating system platform that everything runs on. We assume Microsoft isn't playing games.

Leo: That's a good point. Even if you have open source software, the OS might be involved.

Steve: Yeah, it's ultimately also there.

Leo: Or your Internet service provider. No, I guess it's encrypted by the time it gets there.

Steve: Yeah, but the OS, you're calling APIs to do all this work for you, asking it for things like pseudorandom numbers. Well, what if it generated special Microsoft pseudorandom numbers, and they knew what they were? So again, at some point we have to trust. Unless you go out into the beach and get some silicon in a bucket and bring it back to your garage and start smelting it and purifying it.

Leo: Oh, come on. We don't have to start all the way back. Question 8 from Asher Silberman at Cal State Northridge in California, he's got a Layer 8 club: Hey, Steve. Long-time listener, first-time writer. Started listening to Security Now! back when it started. I was in middle school at the time and listened to it on my walks home. Wow, Asher. Because of you, I was explaining RSA encryption to my friends during gym class. Nerd. But in a good way.

Now I am a sophomore in college and have started a computer security club at my school. We're calling it Layer 8. So far we've entered into competitions and competed in the National Collegiate Cyber Defense Competition - that's cool. I didn't know that existed. That is really cool - and the National Cyber League, even without much practice, since the club just started. So I'm wondering, do you have any tips on resources for learning about security? Any ideas for activities we might do to teach our members over the coming school year? Asher, I'm so blown away. He says: Thank you, your podcast is a great inspiration. But Asher, YOU are a great inspiration. Isn't that a nice email.

Steve: Well, and so I have a neat idea, Leo.

Leo: All right.

Steve: And Asher, here's your project. Something I've been intending to do, but have never gotten around to, is to take a look at all the things we've talked about and all the security technology over the years and come up with a single comprehensive best practices document.

Leo: Oh, wouldn't that be useful.

Steve: And we'll have Asher and his club on the podcast to present it.

Leo: Great assignment. Great assignment. I love it. Best practices.

Steve: Yep, what should people do, how should they behave, what tools make the most sense, easy to use, most comprehensive and so forth. The whole club will have to pull together and answer those questions, distill it down to what people need to do to be secure and what to use. And Asher, I'll get an email back to you with a way to contact me. And we will have you and/or your club, you guys can gather 'round a microphone or appoint a spokesperson, and we'll have you present your results on the podcast.

Leo: And it would be for end-users. You don't have to worry about enterprise or servers or that kind of thing.

Steve: Right. Just for moms and...

Leo: Just end-users running - moms and dads running Windows or Mac, best practices.

Steve: Yeah.

Leo: Love it.

Steve: What to do to stay secure on the 'Net.

Leo: That would be a very nice thing to have. Very handy. And I will publicize it on The Tech Guy, and we'll give you some space on the TWiT site, because I think that would be very useful.

Steve: Yup.

Leo: Love it. Great idea, Steve. Thank you. All right, Asher. Your assignment, should you choose to accept it.

Walter Anthony, in Climax, North Carolina explains how he's cloud-synched tabs in Firefox, just like the cloud-synching of tabs now offered in Chrome v19: I just wanted to let you know I've been using Firefox for a couple of years now in a manner that synchs my home and work PCs. All of my bookmarks, history and tabs are synched between the two. I just run portable Firefox from PortableApps.com in my Dropbox. Oh, that's clever.

Steve: Mm-hmm.

Leo: That's clever. As a result - you know, because Dropbox copies that folder with everything in it - my tabs are always in synch, with history available for each tab. What a great idea.

Steve: Yeah.

Leo: I've also used the synch feature built into Firefox to synch tabs, history, preferences, et cetera, across machines. This has the added feature of allowing my wife and me to both use individual versions of Firefox on the same Windows 7 computer login account. Mine is the portable version; hers is the desktop install. Walt Anthony, U.S. Navy, Retired. That's a great idea.

Steve: Yeah. I wanted to share that. I thought that was really great.

Leo: I love that.

Steve: Yeah.

Leo: And that would work with GDrive and anything that allows you to synch a folder between multiple machines.

Steve: Yeah. PortableApps is a great solution because it's modified the apps so that all of their tendrils are contained within a defined location, basically making the app behave really in a self-contained fashion.

Leo: Great idea.

Steve: So a just beautiful solution.

Leo: Finally, Troy, Asheville, North Carolina, with the Welcome Firefox Add-On Tip of the Week: Steve, I've taken all of your advice since Security Now! #1 and now have moved from Firefox v3.6 to 12, but our beloved Permit Cookie add-on no longer works. So I wanted to give you and your listeners a heads-up that the add-on Cookie Whitelist With Buttons works beautifully, and the same way that Permit Cookies did. Just thought I'd let you know. Keep up the good work. Thank you. So it's called...

Steve: Many...

Leo: Go ahead.

Steve: Yeah, it's called - believe me, that's the whole name, Cookie Whitelist With Buttons.

Leo: Yay.

Steve: And they refer to themselves as CWWB on their page. And I used to be using Permit Cookies. Many of our listeners who were more cookie conscious were using it. And we were all shedding tears and sending notes to each other. It's like, oh, no, it's broken. Permit Cookies no longer works. And sure enough, it died with 11, or maybe before, but I wasn't switching before. But when I finally did, it's like, oh, yeah, doesn't work. And so the good news is Firefox has a cookie whitelist facility built-in, but it's very cumbersome to go and diddle around and poke and navigate through the multilayer UI to get there.

So what these guys have done is they've simply surfaced the things you want to do most with the existing Firefox cookie management onto a couple of toolbar buttons. So just like Permit Cookies, you're able to say "blanket deny cookies," yet when you want cookies for this session only, you can click it. Or if you want to add a site to the permanent cookie permissions whitelist, then you click a different button. So I know that a huge number of listeners who are mourning the passing of Permit Cookies now have an alternative, the Cookie Whitelist With Buttons.

Leo: Woohoo. And that concludes all 10 questions and answers from Steve Gibson. We do these Q&A episodes every other episode, so we'll be doing it again next in a couple of weeks. So you can, if you have a question or a follow-up, go to GRC.com/feedback and ask a question there. GRC is the place where Steve lives. That means that's where you can get SpinRite, the world's best hard drive maintenance and recovery utility. You can also get all of his great freebies, his little apps. And the podcast is there. He makes two special versions available, and they're available only at GRC.com. One is the transcription version we talked about earlier, pure text, searchable. And the other is a 16Kb version. It doesn't sound great [actually it does], but it is small, has the virtue of being tiny.

Now, for the video and the other versions, we have those at our website, TWiT.tv/sn for Security Now! We do this show every Wednesday, 11:00 a.m. Pacific, 2:00 p.m. Eastern time, 1800 UTC, on TWiT.tv. We always welcome the live viewers. The chatroom's very helpful, and it's just nice to have people watching us live. But as I said, we make it available after the fact in a variety of formats, so you can always get it afterwards, too.

Steve: It's hard not to get it.

Leo: Well, no, it's hard to get. But it's not as hard to get as it might be. Or something. Steve, always a pleasure. Thank you so much for your inspiration and your knowledge and for your dietary inspiration, too.

Steve: And we'll be back next week with a next topic. I think I want to talk about this interesting little security problem which surfaced, which is how some firewalls are actually doing a bad thing that allows connection hijacking without being a man in the middle, when they were intending to do a good thing.

Leo: Ooh.

Steve: It's a very clever hack that some smart people have come up with that demonstrates that, once you've got something that's wrong with a protocol, it's really not possible to fix it. And this is a problem with TCP and the fact that sequence numbers are only 32 bits. We've had problems in the past with them only being 32 bits because they were guessable. And some clever people have figured out how to use that fact in a new way to allow connections to be hijacked and intercepted. So a great topic, which we will plow into next week.

Leo: Fantastic. You are fantastic. I thank you, Steve Gibson. I thank everybody for joining us. We'll see you next time, right here on Security Now!.

Steve: Thanks, Leo.