Listener Feedback #175

Description: Steve and Tom 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-422.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-422-lq.mp3

SHOW TEASE: Hey, coming up on Security Now!, Leo Laporte's on vacation, so I get the pleasure of hanging out with Steve Gibson and talking security. Yeah, we're going to talk about the NSA. Yeah, we're going to talk about a zero-day vulnerability from IE. But there is hope in the future, not just from a good television show, but apparently about stronger encryption. All of that and more coming up.

TOM MERRITT: This is Security Now! with Steve Gibson, Episode 422, recorded September 18th, 2013: Your questions, Steve's answers, #175.

It's time for Security Now!, the show that helps you stay safe online. I'm Tom Merritt, filling in for the vacationing Leo Laporte, and very happy to be back doing it with the man himself, Mr. Steve Gibson, the man behind GRC.com, the man I turn to when I want to know what's up in security. ShieldsUP!, SpinRite, been using those things for years. They've saved my bacon. Steve, good to be back on the show with you again.

Steve Gibson: Hey, Tom, you were saying that it was not since last November that we had been doing this, and it was also for a Q&A episode last year.

TOM: That was - right. It was 20 Q&A episodes ago.

Steve: Great to have you back.

TOM: Well, thank you. It's good to be here.

Steve: And has anyone heard from Leo? Did he get there? Is he safe? Has there been any communication? Or has he just disappeared?

TOM: We haven't heard otherwise, so that's good.

Steve: Yeah.

TOM: Yeah, hopefully he doesn't spend too much time. Hopefully he just relaxes and enjoys himself.

Steve: Good, well, that was the whole idea.

TOM: Absolutely. We've got some interesting stuff today, huh, Steve.

Steve: Yeah, not a big news week. It's weird. Sometimes we're like - well, in fact sometimes there's so much to talk about that we just, like, it pushes any other end-of-show content off the end. I've got some interesting stuff always, and some great comments and thoughts, a couple really long pieces. But they were so interesting, I thought, well, we'll have time. So, yeah, I think we'll have a good show.

TOM: There's always something to say about the NSA these days. So we've got something about...

Steve: Ah, yes.

TOM: And thankfully, well, not thankfully, but for our purposes there is a zero-day vulnerability for IE we can talk about. Thank you, hackers, for doing that. All right, Steve. Let's start off talking about that zero-day vulnerability. Is it for everyone? How bad is it?

Steve: Well, it's bad enough that anyone - the way I tweeted it this morning when - I got email from Microsoft last night, or I guess actually late in the morning yesterday, and finally got around to checking it out. So what I said was, in my tweet, "New IE 0-day Vulnerability being exploited in the wild. If you must use IE you can apply temporary Fix-It," and then I gave a little bit.ly link [bit.ly/1gyQ31T]. So anyone who's interested, if you check my Twitter feed, Twitter.com/SGgrc, and you'll see my most recent stuff. There is a Fixit which they describe as a "shim," which will solve the problem. They're reporting...

TOM: Not a shiv, a shim.

Steve: Is it shiv?

TOM: No, no. That would be the opposite, I think.

Steve: Yeah, the vulnerability would be a shiv.

TOM: That's right. The shim protects you from the shiv.

Steve: Yeah. So they're seeing, Microsoft has acknowledged the exploitation of IE8 and 9, although this does affect all versions of IE, 6 through - and they even list 11, even though it's like, not out of the box yet.

TOM: Oh, wow. Yeah, that's just a preview.

Steve: So I love this Microsoft speak, the way they write these things. I mean, clearly it's boilerplate. But of this they wrote: "Microsoft is investigating public reports of a vulnerability in all supported versions of Internet Explorer." Is IE6 still supported?

TOM: I didn't think it was.

Steve: I don't think it is anymore. But anyway, it is on their list of, like, vulnerable OSes.

TOM: They know so many people still use it, they probably just put it on there, yeah.

Steve: All vulnerable products. Anyway, so they said: "Microsoft is aware of targeted attacks that attempt to exploit this vulnerability in Internet Explorer 8 and Internet Explorer 9. Applying the Microsoft Fix it solution" - and they give the CVE number, so it's MSHTML, it's an MSHTML code, MSHTML Shim Workaround - "prevents the exploitation of this issue. See the Suggested Actions section of this advisory for more information. The vulnerability is a remote code execution vulnerability. The vulnerability exists in the way that Internet Explorer accesses an object in memory that has been deleted or has not been properly allocated. The vulnerability may corrupt memory in a way that could allow an attacker to execute arbitrary code" - again, I get a chuckle out of this because they previously said it does allow an attacker to do this, and attackers are doing this.

Anyway, so they said, "in the context of the current user within Internet Explorer. An attacker could host a specially crafted website that is designed to exploit this vulnerability through Internet Explorer and then convince a user to view the website. On completion of this investigation, Microsoft will take the appropriate action to protect our customers..."

TOM: Thank you.

Steve: As opposed to not having written the code in the first place, which would have always had them protected. "Which may include," they continue, "providing a solution through our monthly security update release process, or an out-of-cycle security update, depending on customer needs." So we don't have - you can't go to Windows Update for this. So you do have to go get this deliberately, a Fixit solution, the little "1," you know, click on the "1" button, and then it turns something off in the registry or installs a quick patch or something.

So, again, if you're - when I tweeted this I got some responses from people saying that they had to use IE in their corporate settings, so they appreciated the heads-up. They had passed this on to IT to get the IT blessing before they did it, which is what you should do in a corporate setting. And if for some reason you're still using it at home - that is, IE - or on your own, then it's probably good to do. It's funny, too, because they have, like, mitigating factors that they list. It's like, okay, things that make this not a problem. And the first one is, like, the Server versions of IE use their so-called "protected mode"? Well, you might as well just unplug from the Internet if you try to use IE in that mode. When I...

TOM: It's kind of for browsing manual files, isn't it? I mean, yeah.

Steve: It's incredible how restrictive it is. I can't figure out what it's useful for, and obviously you are protected when you do that. The fourth mitigating factor, though, the last one, they said, "In a web-based attack scenario, an attacker could host a website that contains a webpage that is used to exploit this vulnerability. In addition, compromised websites and websites that accept or host user-provided content or advertisements" - okay, well, what website doesn't have user-provided content or advertisements?

TOM: These days.

Steve: And that was my point, is that...

TOM: Yeah, Geocities is gone, so, yeah.

Steve: Yeah. What we're seeing is that the way these exploits are getting to people now is not necessarily that you go to some dark corner, off-the-path site where you probably ought to know better. It's you're at The New York Times or the Wall Street Journal or some Yahoo! page that you would expect to be perfectly safe. And somehow that page has been made vulnerable to exploitation, and something gets stuck on the page. So there's really, you know, it's not like if you are surfing safely, you're safe now. The problem is, if you're surfing with IE, you're more than likely not safe.

TOM: Yeah. One of the fixes for me is to not use IE unless I have to. But some people have to; right?

Steve: Yeah. I did note in the news coverage today of Google's new tracking alternative to third-party cookies, AdID, it think they call it?

TOM: Yeah.

Steve: I didn't realize that Chrome is now the No. 1 browser and has surpassed IE and Firefox and the others.

TOM: Depends on who you ask. But in lots of those surveys they have, that's true.

Steve: I just wish it were smaller, I mean, lighter weight. I've talked to Leo about this. When I launch Firefox, I get a bump in memory. When I launch Chrome, I, like, lose a quarter of my machine.

TOM: It used to be the opposite. The reason I switched to Chrome years ago was because Firefox was so big back then. So I guess when you become popular you start to become a memory hog. But the thing about Firefox and Chrome, I use NoScript and NotScript in them, which would help prevent this sort of thing from happening. But is there a NoScript equivalent for IE where you can say, like, turn off all the scripts on these websites so that third-party stuff doesn't execute?

Steve: You can absolutely disable scripting globally. But I've never seen an add-on that allows you to do it dynamically. It may be that the add-on interface doesn't provide the hooks necessary in order to do that. But, boy, I mean, that would be a great solution. But the only thing I know if is, like, disable JavaScript completely. And then, unfortunately, too much of the web is broken.

TOM: Because the brilliance of NoScript is I can say, oh, yeah, I'm on The New York Times. Let The New York Times scripts work so that the site works well. But don't let any of that other stuff from outside run because I don't know what that is.

Steve: Yeah, and in fact in whatever they're on now, I think it's 23, 22 or 23 of Firefox, we were commenting a couple weeks ago that they had somewhat controversially removed the setting from the UI to allow users to disable JavaScript. And when you updated, because it might cause so much trouble, they silently reenabled it as they were removing the checkbox from the UI. And so Leo and I discussed it, and I was curious how they thought they could get away with this. So I pursued the dialogue, like down in the forums where this was argued, and their position was scripting is so necessary now that only experts know not to run with it or to run with it conditionally. And anybody who

wants conditional scripting needs to use NoScript. So they're sort of like - they were, like, taking themselves out of this all-or-nothing mode, very much the way IE has, actually. But at least Firefox has NoScript, and Google has NotScript that gives you per-site control back, which is useful.

TOM: This next story we talked about on TNT. And we had a lot of differing opinions about...

Steve: Yeah, it's interesting. I had the TV on, I think it might have been Monday morning. And it's like I was watching "Morning Joe" on MSNBC just sort of babbling on in the background. But this, so it was like, just yanked my attention when Joe was holding up the L.A. Times with a story because I thought, whoa. Maybe this is going overboard. So the L.A. Times covered this. The headline was "Glendale School District to Monitor Students' Social Media Posts." Kelly Corrigan, reporter for the L.A. Times, wrote: "Glendale school officials have hired a Hermosa Beach company to monitor and analyze public social media posts, saying the service will help them step in when students are in danger of harming themselves or others.

"After collecting information from students' posts on social media platforms such as Facebook, Instagram, YouTube and Twitter, Geo Listening" - which is the name of this Hermosa Beach company - "will provide Glendale school officials with a daily report that categorizes posts by their frequency and how they relate to cyber-bullying, harm, hate, despair, substance abuse, vandalism and truancy. Glendale Unified, which piloted the service at Hoover, Glendale and Crescenta Valley high schools last year, pays the company $40,500 to monitor posts made by about 13,000 middle school and high school students at eight Glendale schools.

"According to a district-wide report, Geo Listening gives school officials 'critical information as early as possible,' allowing school employees 'to disrupt negative pathways and make any intervention more effective.'"

TOM: Wrong thinking? Is that - yeah.

Steve: "Glendale Unified Superintendent Dick Sheehan said the service gives the district another opportunity to 'go above and beyond' when dealing with students' safety. 'People are always looking to see what we're doing to ensure that their kids are safe. This just gives us another opportunity to ensure the kids are safe at all times'" - whether they're at school or not, I add that - "he said. Yalda Uhls, a researcher at the Children's Digital Media Center at UCLA and a parent of two, said students should be made aware that their posts are being monitored. 'As a parent, I find it very big brother-ish,' Uhls said, adding that students could lose trust in adults once they find out their posts are being tracked.

"However, she also admires schools' efforts in trying to attack the problem of cyber-bullying. 'This could be one piece in a school's tool kit to combat that problem, and it should be a very small piece,' she said. School board member Christine Walters said that as Glendale educators have become increasingly aware of how much bullying occurs online, officials have become more 'proactive to find ways to protect our students from ongoing harm,' she said. 'Similar to other safety measures we employ at our schools, we want to identify when our students are engaged in harmful behavior.'"

TOM: Right. That's why the schools are always putting microphones up and recording all the conversations in the halls, and they're following students home to see where they go in public. Those are sort of the analogs I came up with when I heard the story because it is important to realize that they're not spying on the students by, like, finding their

emails or anything. They're looking at public posts. If their Facebook posts are private, then they're not going to see them. But the Twitter posts are always public. So there isn't the invasion of privacy that you might think of. But there are other things that students do publicly that schools don't try to monitor.

Steve: And thinking in terms of execution I wonder if there is then - is there a form that students fill out of the account names? Like what's your Twitter handle? What's your YouTube account? What's your Instagram account? What's your Facebook?

TOM: Oh, that's interesting, yeah. How do they find out who the students are? Yeah.

Steve: Yeah. I mean, so, and, I mean, I completely agree that this ought to be done with students' awareness that their school is listening. On the other hand, the way actual social media works, I'm sure this thing has been known from the second it was deployed that your school is monitoring what you're doing. So my sense is all this does is push that kind of stuff further underground, if somebody wants to communicate that way. I don't know.

TOM: Well, in a way I could almost see it as a positive in that it teaches kids, hey, everything you do on the Internet's public; right?

Steve: Yes. I had the exact same...

TOM: If you don't want people to know about it, be careful.

Steve: Yup, exactly. It's like, hey, get used to the new connected world, kiddies, because everyone is watching what you post. And there has been a lot of dialogue about the consequences of oversharing on Facebook, the fact that now headhunters and employment agencies and employers are very much doing a deep dig into what would-be employees have posted publicly in order to get a better sense. The reading is you can better gauge who somebody is from that than sitting across from them at a table and asking them a series of canned questions for which they're going to look like an angel.

TOM: It is a really interesting debate because of the fact that they're public posts; right? And everyone's reacting as if they're spying on the kids' private interactions.

Steve: Yeah, you're right.

TOM: And they're not. And I'm not saying that I think it's a good idea, either. I think it does go over the line. It's a little bit overreaching. It's a little bit of a nanny state type of thing, if some people like to categorize it that way, because you can't protect kids from every kind of harm. But at the same time, children are doing these things in public. Anybody can see this stuff.

Steve: And I guess you could also argue that this is - the proper role may be parents, but parents aren't doing this. So the school, sort of being a little bit of a nanny state mode, is stepping in to take responsibility. And it's interesting that there is a big service that is making $40,000 a year, no doubt automating this in some way and doing, like, keyword searches. I mean, certainly there aren't any - there's no one who's reading 13,000 students' individual, you know, every posting everywhere. So this is like a small version of what the NSA is doing on a local scale.

TOM: I kind of wish Oracle was a little more of a nanny state, to be honest.

Steve: [Laughing] Yeah. This is kind of creepy. We've talked about the fact that Oracle, that Java 6 is no longer being updated, and the problem that represents. But exploring the logical consequences of that a little bit further is extra chilling. Dan Goodin, writing for Ars Technica, reported on some studies, I think it was Trend Micro that sort of brought this to his attention. He wrote: "The security of Oracle's Java software framework, installed on some three billion devices" - which of course it brags every time you update it, we used to see that all the time when we were updating it frequently - "is taking a turn for the worse, thanks to an uptick in attacks targeting vulnerabilities that will never be patched" - and that's the key, so we're seeing an uptick in attacks on vulnerabilities that will never be patched - "and increasingly sophisticated exploits. The most visible sign of deterioration is in-the-wild attacks exploiting unpatched vulnerabilities in Java version 6," said Christopher Budd, threat communications manager, oh, yeah, at antivirus provider Trend Micro.

"The version, which Oracle stopped supporting in February" - that is, the whole line of v6 - "is still used by about half of the Java user base. Malware developers have responded by reverse" - and here's the key, sort of like extensions of what you might expect actually happening. "Malware developers have responded by reverse-engineering security patches issued for Java 7 and using the insights to craft exploits for the older version. Because Java 6 is no longer supported, those same flaws will never be fixed." So, Budd adds, "This is a large pool of vulnerable users who will never be protected with security fixes, so [they're] viable targets for attack." Anyway, so...

TOM: Yeah, I mean, just don't use Java if you can help it.

Steve: Yeah, exactly. Don't use it unless you have to.

TOM: And some people have to. That's the problem.

Steve: Yeah. And if you do - see, now, the problem is that many corporations have old software that won't run on Java 7. So their employees and their systems are stuck with Java 6. They look at the burden of updating the software to run on 7, and it's like, uh, we have other things to do. It works. Let's leave it alone. And the problem is it's just - it's waiting to be exploited. So hopefully those are non-browser-hosted instances. Or, I mean, if a company wanted to be secure, they could install Java that they have to use on IE, but don't let IE talk to the Internet. Just use Java as an application runtime locally, and host it in a browser if it has to be. But then on your browser that you use for the Internet, Chrome or Firefox, for example, there you want to use Java 7 or no Java at all. Probably no Java at all. I can't really think of any contemporary site that requires that you have Java itself installed.

TOM: No, it's usually an application, yeah.

Steve: I guess I've heard - whenever I say this I get tweets from people in the Scandinavian countries saying, oh, my bank requires that I use Java. But then I'm also seeing that they're rapidly moving away from that. They're probably, unfortunately, going to JavaScript. But that's better than Java.

TOM: And so you're saying browse globally, execute Java locally.

Steve: Yeah. Well, yes. Because the great benefit of Java was its platform neutrality. You could run Java - a corporation could write some proprietary system, maybe it's for access to their backend servers or who knows what, some big glue that pulls things together. And it would run on Macs, and it would run on PCs, because Java was the virtual

machine, the so-called JVM. You'd install that on whatever computer you wanted, and then that one application, it was write once, run everywhere. So that was a reason that Java had so much uptake sort of early on. But that's not a browser-based approach. In my opinion, the whole notion of making Java run in a browser was, in retrospect, which obviously is hindsight and easy to do, it was a disaster because it's caused so much grief for people. So it made sense to run it as a local virtual machine in a write once, run everywhere mode. Never was it a good idea as a browser plug-in because it's just been a catastrophe.

TOM: It is kind of sad. I wish it had been - I wish it had fulfilled the write once, run everywhere. But we kind of moved past that with HTML5 and web apps and such.

Steve: Exactly. I think we're really seeing - well, in fact, for example, we're even seeing now, with the new communications stuff, we're beginning to see like real-time audio and video leaving individual chat apps and moving into the browser. I think the browser as a dynamic container clearly is getting powerful enough to write and to run these kinds of things.

TOM: Okay. It's NSA time. We need kind of a regular bumper or radio stinger for this, huh.

Steve: Yeah, yeah. We can't get away from these things. And there are a couple interesting notes from our users that we'll share here in the second half of today's podcast. But this one sort of caught my eye because the person who was quoted, John Gilmore, is a famous co-founder of the EFF, very involved in what's happening on the Internet. And this was in Infosecurity magazine. The title of this piece was "Did the NSA subvert the security of IPv6?" And I'm just going to share this little piece, it's pretty short, in its entirety.

TOM: That's frightening, yeah.

Steve: They wrote: "Following the Snowden leaks revealing Bullrun," which we talked about last week, the whole notion of the extent and the level and the budget of the NSA's overt - now it's overt, no longer covert - intent to essentially keep the Internet from going dark for them, "there is an emerging consensus that users can no longer automatically trust security. Cryptographer and EFF board member Bruce Schneier" - who's been a recent guest of Leo's, and of course we speak of Bruce all the time - "has given advice on how to be as secure as possible. 'Trust the math,' he says. 'Encryption is your friend. Use it well, and do your best to ensure that nothing can compromise it. That's how you can remain secure even in the face of the NSA.'" Unquote from Bruce.

TOM: I like that "trust the math."

Steve: Trust the math.

TOM: I like that as a slogan, yeah.

Steve: Yes. Because he's absolutely right. That, and, I mean, as far as anyone knows, that is still the case. The actual math, the actual technology of cryptography is sound. Unfortunately, the actual delivery of security uses cryptography as only one component. And nothing else except the math is sound. And so the NSA, they can bribe people. They can use national security letters. So they can use arguably abusive authority and basically everything but the actual crypto. The crypto we can depend upon. So the story continues.

"He confirms the growing consensus that Bullrun's greatest success is in subverting the implementations of encryption rather than in the ability to crack the encryption algorithms themselves. The general belief is that the NSA has persuaded, forced, or possibly even tricked individual companies into building weaknesses or backdoors into their products that can be exploited later." And I've gone on the record to say, okay, I need evidence. It's like, yes, we need to be skeptical because we've always had to be skeptical. And I think what has happened in the wake of these revelations is that the tinfoil factor has gone up by an order of magnitude. But it was always there, so...

TOM: Yeah. In a way, the NSA is just a very large and powerful group of hackers. And so you need, I mean, they're big; right? And there's political implications. But as far as, like, what they can do, they can do all the same things anybody else can.

Steve: Yeah. Now, and we talked for example about BitLocker, Microsoft's proprietary hard drive encryption. I'm not going to use it. I'm going to use TrueCrypt because I know that TrueCrypt came from an international group of developers where the security of the product was their goal. I don't know anything about Microsoft's goals for BitLocker, nor any influence. I have no reason to believe there was any. I actually, again, I would need proof rather than just this sort of, like, conjecture and suspicion. But still, given a choice, hey, we have a choice. TrueCrypt is free. Why not use that one?

Anyway, continuing: "The bottom line, however, is that the fabric of the Internet can no longer be trusted. Meanwhile, John Gilmore, co-founder of EFF and a proponent of free open source software, has raised a tricky question: Has NSA involvement in IPv6 and IPSEC" - the IP security layer - "effectively downgraded its security? IPSEC is the technology that would make IP communications secure." So just let's stop again here for a second to remind people.

The way we get security over TCP connections now is we establish a point-to-point connection, and then we run a protocol SSL/TLS on top of that lower level physical connection, or point-to-point connection. So HTTPS, for example, brings SSL to the HTTP connection as an add-on, as a layer on top of the underlying protocol. One of the best things, in my opinion, about IPv6, which almost makes it worthwhile, was that IPSEC, which again has been an existing technology, IPSEC has been around for a long time, you can get IPSEC routers, you can set up tunnels and so forth. So that's been all in place. But the proposal has been that it would be integrated into IPv6 from the beginning. So that was one of the cool advances was that IPv6 would have security as part of, like, bound into the protocol. IPv4, that we're using now, doesn't. You need to add that layer. IPv6 would.

So John says: "Gilmore notes that he had been involved in trying to make IPSEC 'so usable that it would be used by default throughout the Internet.' But 'NSA employees participated throughout the process and occupied leadership roles in the committees and among the editors of the documents.' The result was 'so complex that every real cryptographer who tried to analyze it threw their hands up and said, "We can't even begin to evaluate its security unless you simplify it radically."'"

TOM: Hmm, that's worrying.

Steve: Which never happened. So, I mean, so this, everything we know, this reads true because, if you wanted to kill something, you'd just keep adding crap to it until it is so huge and lumbering and burdensome that, first of all, nobody wants to code it because it's just a nightmare to, like, support it all. And with that complexity, as we know, complexity is the enemy of security. So you can imagine - and again, pure speculation.

But even if hidden in that complexity wasn't some way around it, the point was that real cryptographers understood that, if they couldn't understand it, they couldn't ever vouch for its security. So even if it was secure, you just make it impossible to implement, and nobody will.

So continuing, Gilmore says, or the article says: "Gilmore doesn't explicitly say that the NSA sabotaged IPSEC, but the fact remains that in December 2011, IPSEC in IPv6 was downgraded from 'must include' to 'should include.' He does, however, make very clear his belief in NSA involvement in other security standards. Discussing cell phone encryption, he says, 'NSA employees explicitly lied to standards committees' leading to 'encryption designed by a clueless Motorola employee.'"

TOM: Poor employee.

Steve: "To this day, he adds, 'no mobile telephone standards committee has considered or adopted any end-to-end, phone-to-phone privacy protocol. This is because the big companies involved, the huge telcos, are all in bed with NSA to make damn sure,'" the article says, "'that working end-to-end encryption never becomes the default on mobile phones.'"

TOM: And this makes perfect sense when you think of all those stories we've heard over the years of somebody tapping in on cell phone conversations, showing how easy it is to listen in on cell phone conversations, and everyone reacting, going why don't they do something about this? Why don't they fix it? And the best cover story of all was that we're like, agh, the telcos, they just don't care, they're greedy, and they're incompetent. And probably several people out there jumped to the conclusion that it was some kind of cooperation with the government. But that seems 100% the most likely explanation now.

Steve: Yeah, unfortunately. And again, that demonstrates this has been going on for quite a while. I mean, and again, the NSA's charter is to know what's going on to the greatest degree possible. So it is the case that our lives have moved online, what we do is moving over wires, and so I used to buy things driving around in my car. Now I go to Amazon to do it. And so if somebody, I mean, so Amazon has all my records. Amazon knows more about me than any other single entity on the planet, based on what I buy.

So, I mean, the world has changed as we've gone online. And I think what's interesting is that it probably got really good for a while for the NSA. But I've said on the podcast, talking to Leo about the Snowden effect, that it's going to get really bad in the future. I mean, this will end up having been the worst thing that has ever happened to our intelligence collecting ability because, as Bruce says, trust the math. The math is good. And so the point is that the math will stay good, but people who care about offering more security have the ability to do so, in all the ways other than the math. And that's there. So, which is a long-winded way of saying we're going to fix a lot of this.

TOM: Yeah.

Steve: And it's going to be because we now know somebody is actually listening, and we'd rather not be listened to.

TOM: The implications of this are not going to be known in full, any more than the Pentagon Papers coming out at the time did we know in full what that was going to do to how we conduct business and how we think of privacy and how we think of the government. It's still being worked out right now. But for sure it's going to keep coming. It does seem like there's more leaks to come, too. I don't know if we're at the end of

them yet, or not.

Steve: Yeah.

TOM: Let's cheer up. Let's talk about "Orphan Black."

Steve: Well, I just wanted to say I mentioned it a couple weeks ago. A good friend of mine sent me a review from Paul Krugman, of all people, a well-known columnist for The New York Times, who raved about it. I've still not seen it. In fact, oh, here it is, thanks to Amazon.

TOM: Wow, that was quick.

Steve: There's my "Orphan Black" DVD. I'm still catching up on "Mad Men," having figured, okay, everyone's raving about it, I need to find out what's going on, after catching up with "Breaking Bad." So I'll get to "Orphan Black." But I'm getting a ton of feedback from people making me very glad that, even sight unseen, I shared this with our listeners. Will Pearce, and I saw this in the mailbag I was going through today for today's Q&A, Will Pearce in Raleigh, North Carolina had the subject line "Orphan Black," so that just kind of caught my eye. He just said, very quickly, two lines, "Addicting and bingeing." Or, I'm sorry, he said, "Addicted and bingeing. Thanks a lot, Steve. I really needed another time sink." Which is obviously a backhanded compliment.

Then also Dale Francisco in Fresno, California, subject was "Orphan Black Nightmares." He said, "Steve, based on your critique," which actually wasn't that, it was not even a review, just awareness, "in Episode 420, I decided to see what about 'Orphan Black' could be so special. Dang it. I usually watch the History Channel, Discovery, Military Channel, and PBS. Argh. Now I will have to burn up more hours of my day with 'Orphan Black.' I fully expect to spend nearly all day tomorrow watching/catching up with 'Orphan Black.' Thankfully, I just retired. I have a few more hours each day available to devote to my personal interests."

So again, for what it's worth, I've still not seen it; but the people who, our fellow listeners who have, have been raving about it. I haven't seen a single negative comment.

TOM: Oh, yeah, no. I can back them up. I watched the entire season on BBC America.

Steve: Oh, cool.

TOM: And it was fascinating. And they're airing it on the BBC in the U.K. now because it was actually made over here.

Steve: Oh, so I know that they're rerunning first season. Are they...

TOM: On BBC America they're rerunning first season. In the U.K. they're getting it for the first time.

Steve: Ooh, which means people who have [clearing throat] ways can no doubt get it, too.

TOM: Oh, of course, yeah, absolutely. And I'm excited about the second season and what they're going to do there. We're going to meet more clones. I don't want to say too much, if you haven't watched it yet. But, yeah, it's good.

Steve: Cool. So a little SpinRite update, and then we'll get into our Q&A. Yesterday I finished the work that was, like, the whole first phase of development. The major feature that SpinRite 6.1 will offer is freedom from the BIOS. The BIOS has been an increasing problem for SpinRite.

TOM: That figures. That's great.

Steve: Unfortunately, yeah, the BIOS hasn't been able to keep up with the size of drives. And so sometimes people report divide-by-zero errors when they give it - when a particular motherboard and a particular drive don't like each other, SpinRite comes along and says, oh, let's go, and the BIOS collapses. So what we have now is a rock-solid implementation, our own direct memory access, bus mastering, native-level drivers for every system that the hundred people in the newsgroup, I think there's 175 drives I saw. Somebody pulled the information together and posted a table. It's really a fabulous development environment to work with these guys. And it's working perfectly. It's really interesting, too, because what I ended up creating was a benchmark which is perfect.

SpinRite now has, or I should say - it's funny, too, because I have to be careful because people have been writing to GRC saying, hey, where's the beta that Steve's talking about? We're not - I don't have a SpinRite beta. Actually we have something called SpinTest, which is sort of the core, the new core communications structure, which I will then merge into SpinRite once that's finished. So it's a freestanding application that anyone who is interested with could play with, and it's freely downloadable. What we're seeing is - oh, I'm sorry, what I was going to say was that I ended up producing a benchmark just because that was a nice way to demonstrate what my goal was, was performance.

So SpinRite has an integrated extended memory manager that we created from scratch. I talked about how in real mode I'm able to tweak the Intel architecture to get access to 4GB of memory. We only really need 32MB. 32MB is the maximum size that a disk could transfer at once. That's 16 bits' worth of sector count, which is 65536, which is 32MB of data. So what we're seeing is we're able to demonstrate data transfer rates that match the specs from the companies making the drives. Obviously, they're going to quote the highest possible data rate their drive can achieve. This benchmark gets the same number. And, for example, on the faster drives that we're seeing, we're getting, like we're seeing 157MB - I think it's 57. I didn't go back and check. It's either 57 or 75. 157, something like that, MB/sec, which equates to, in terms of like a SpinRite Level 2 read and recover when necessary scan, 93 minutes per terabyte.

So we've really achieved what I was hoping for, was the ability to test large drives in a short amount of time. 93 minutes per terabyte means a little over an hour and a half per terabyte. So you can do 1, 2, and 3TB drives just in a matter of a couple hours. So it's going to be - so at this point the technology is in place for the older style standard, the ATA IDE standard. There's a newer type of controller, so-called AHCI, the Advanced Host Controller Interface. That I'm going to do next. But first I'm going to take a break, as I've mentioned before, to document the idea that I had for website authentication. I think I have a way of obsoleting usernames and passwords completely. I've been thinking about it in the background for a couple weeks, since this hit me on a Thursday morning during breakfast. I can't find a problem with it. So I'm going to - so I wrapped up SpinRite work yesterday. I'm going to get this concept for authentication documented. Then I'll talk about it here on the podcast, and then I'm going to get back to SpinRite.

TOM: That's impressive stuff, Steve, that being able to operate outside of the BIOS. And I for one am one of the many who hope you fix that username/password thing, too. That would be amazing.

Steve: Yeah. It's funny because, well, I've seen people talk about like the additional complexity of one-time passwords and how the problem with its adoption is that it hasn't removed anything. It's added more complexity in return for the assumption of more security. And but the problem is - so there's a problem because you're just - you're adding more stuff. And yes, okay, maybe it makes you more secure. I mean, obviously I've been a fan of one-time passwords. If that's all you can do, it makes you more secure. I think I have a solution which completely replaces the whole username password model. It can run side by side. And so a website could offer traditional authentication or new style. And the motivation for new style would be it's much easier to use. It's, for example, safe in a public setting. You could safely log on in a library, where a system might be crawling with malware. And the reason it would be, one of the many reasons it might get adopted is the website then doesn't have the problem of user credentials escaping from them, either. It's safe against that, as well.

So anyway, I don't mean to keep teasing people. I've been teasing myself for weeks because I haven't let myself sit down and really focus on it until I finished this phase of SpinRite, which as of yesterday is done.

TOM: It sounds great. And people in the chatroom are like, ask him this, ask him that. I'm like, let's let him document it, get the ideas down. Then we can start pressing him for details.

Steve: Yup. And I want it attacked. I absolutely need more eyes on it in order to say, what about this, and what about that.

TOM: Yeah, for sure.

Steve: So we'll be doing that for sure.

TOM: Time now for a Listener-Driven Potpourri #175. I love doing these. These are fun, Steve.

Steve: Well, we got a bunch of really interesting thoughts and observations and questions. So I think we're going to have fun with them.

TOM: Let's start off with listener Greg, writing from an undisclosed location because he's probably protecting his geolocation. He wonders what Bruce Schneier meant: Hi, Steve. Could you briefly explain what Bruce Schneier meant when he said that he preferred conventional discrete-log-based systems over elliptic-curve systems because the latter, referring to elliptic-curve, have constants that the NSA influences when they can.

Steve: Yeah. I'm a little concerned that elliptic-curve systems are going to be - are going to have their reputations needlessly damaged by being lumped in, as with, like, all elliptic-curve systems. The point is that, where the traditional RSA-style public key technology, as we've talked about often, uses the difficulty of factoring a really large number into the two primes of which it is the product. So there it's, like, very easy to understand what that does. A whole different class of problems, where factoring is the hard thing that no one knows how to do quickly. The so-called discrete-log problem, is the fact that we don't know how, similarly, to find the discrete log of a large number.

Now, there are different systems that use the logarithm problem, one being elliptic curves. Elliptic curves are an equation, x^3 equals x^2 plus x times 3 plus b or something. I mean, so it's basically an algebraic curve. But it is parametric in nature. The specific elliptic curve that you choose bears on the performance. So, and cryptographers

understand that there are dumb ones that no one would use, and then there are lots of good ones that people can use. So what's happened is - but inherently there are, like, curve families. You are able to plug these variables in and choose an elliptic curve. So for interoperability of systems, in the same way that we have, like, clients and servers that need to be able to have an agreed-upon protocol for establishing a handshake, you'd like to agree on specific elliptic curve parameters so that you could then write protocols that use those curves.

So here's where Bruce's concern and the NSA influence comes in. We talked about this a little bit last week with regard to a wacky random number generator that the NSA had a hand apparently in influencing. Specific elliptic curves have been standardized by the NIST, and the parameters are known. The problem is we don't know where the parameters came from in many cases. And it's been shown that it's theoretically possible to choose parameters that effectively give a specific curve a backdoor. So essentially, if you know where the parameters come from, if you know who created them and why they created them, then an elliptic curve is not only as good as RSA - and this is the point - it's arguably much better.

The reason people are looking at this, and in fact considering elliptic curves and discrete-log problems as the generation beyond RSA, is that, from everything that we know, the discrete log problem is much more difficult to solve for a given size of problem, for a given bit length, than factoring. So, for example, we know how to - say that you had a number that was small, like it was no larger than 50 bits. Well, we know how to factor 50-bit numbers. I mean, that's easy to do. But doing a discrete log of 50 bits, even that is much harder than factoring. So the point is that equal difficulty, discrete logs versus factoring, the discrete-log problem is much harder at a given bit length. Consequently, you can use much smaller keys in elliptic curve discrete-log systems than in the RSA factoring problem system, and much smaller keys means much faster algorithms.

So in terms of difficulty, elliptic curve systems appear to have a much stronger future because they are vastly faster today for the same level of security. And it looks like they will scale very well in the future as we decide we need more security by creating greater bit lengths. So there's nothing wrong with them at all. I'm a fan of elliptic curve systems. You just have to understand why you're using the curve you're using. And I agree, unfortunately, all the standard curves are, I think, poison at this point. The NIST standard where, like, these are the curves we're all going to use, it's like, uh, no, thank you.

TOM: So in other words, trust the math, but not the mathematician.

Steve: Well, again, it's like, unfortunately this particular system, a prime factorization is not parametric. There's no parameters to be used at all in factoring a number. There's the number. Factor it. That's all you can do. Elliptic curves, by their nature, are a family of curves. There's an infinite number of them. And so somebody chose the particular parameters for a specific curve that you could then agree to use for your system. The question is, who chose those, and why?

TOM: Yeah, what family? Whose family? And everything we've been saying, which is there's nothing wrong with the math of elliptic curves, but there's these endpoints of, like, here, use this set, use this family. Who's saying?

Steve: We'd like you to standardize on this. It's like, uh...

TOM: Yeah, exactly. Richard Warriner in Bedford, U.K. offers some sci-fi feedback: Steve, I just wanted to thank you for introducing me to the Antares trilogy by Michael McCollum.

It has been some time since I have read a complete trilogy back to back and couldn't put it down. For me, it has just the right balance between the sci-fi element and the main narrative of the lives of the characters - something that I struggle with sometimes with Peter Hamilton books. I certainly won't be able to wait many squared heartbeats before I read another of his books. Richard.

Steve: So I just wanted to, you know, we've talked about Michael McCollum often. And I wanted to say to Richard, well, if you liked the Antares trilogy, don't forget the Gibraltar trilogy. There's another trilogy: Gibraltar Earth, Gibraltar Sun, Gibraltar Stars. Same author, and I recommend it similarly without reservation. And to any of our listeners who have heard me talk about this in the past, but haven't made the move, if your life gives you some time to read, either the Antares trilogy or the Gibraltar trilogy by Michael McCollum are fabulous. That's M-c-C-o-l-l-u-m.

TOM: And mentioning Peter Hamilton, I'm reading "The Great North Road," which is his latest one right now.

Steve: Cool. I've not yet started.

TOM: Definitely a balance of characters and sci-fi in that one going on. Which I know what Richard's talking about sometimes. But this one I think has a really good balance. So that's another one.

Steve: Cool. How far are you?

TOM: About, well, it feels like I should be at the end, but I think I'm about a quarter of the way through, yeah. They're long.

Steve: They are.

TOM: Daniel in Oslo wonders whether we're living in the post-encryption world already. He says: I guess I've lost faith. With the latest revelations that the NSA can crack pretty much anything on the market these days, I am even skeptical of the news today of breakthroughs in quantum encryption. Is this a new attempt at sneaking in backdoors and getting everyone to jump over to a new, even more easily circumvented standard? I seriously considered unplugging myself entirely for the first time today. Is there any hope, Steve?

Steve: So this question and the next one are pretty much on this topic. I guess what I want to reiterate is that I believe there is hope. I think that, as Bruce says, trust the math. We also, as we've seen, have to trust the system. And it's the system, not the encryption, not the math, which has, because it is a system, and it's complex, and it involves lots of moving pieces, we realize now it can be abused. The system can be tightened up. The system can be fixed. And I think what's going to happen, a year from now, this doesn't happen fast, but there is tremendous pressure now on improving the system.

And that's why I think that ultimately what happened with Edward Snowden is going to end up really improving security because now we know there's a reason to tighten things up, and there's a reason, for example, to update protocols. Protocols are not easy to update. There's inertia, huge inertia to updating them. And so it's like, well, they seem to be fine right now. Well, no one thinks that today. So they can be updated. They can be fixed. And I really think we're going to see a technical response which is going to go a long way to bringing back the faith that Daniel has lost. It's worth - it's going to take a

while, but I believe it's going to happen.

TOM: Well, and things, when you take the historical perspective, are better now than they have been. And I'm saying the historical perspective. I'm not talking about last year or the year before or even 10 years ago. But people actually have the ability to buy a computer and use encryption and not just slave away in a field somewhere for a subsistence agriculture, at least in large parts of the world they do.

Steve: And look at the appreciation that really has - it took a long time. But people now understand not to use "password" as their password.

TOM: Right. Even on a smaller scale, that's a big advance, you're right.

Steve: Yeah. It was inertia. It took a long time. But people get it now. So this kind of change isn't instantaneous. I would say we're at the low ebb at this instant because there hasn't been time to react yet, technically. Yet at the same time we've been hit by the political, the social side. The technology's going to come.

TOM: Yeah. We actually have a concept of privacy that has not always existed. In fact, that's a very new thing in human history. And so we have to work it out. And this is it being worked out. This is the sausage being made. At least that's my opinion.

We've got a similar one here. Lynwood Wright in Tampa, Florida wonders whether privacy exists at all: Steve, I've been an IT professional for 17 years, longtime listener, and longtime user of SpinRite. I can't count how many times it's recovered data. Genius product.

I've been following all of our government domestic spying news, and I can't think of any possible way to maintain privacy, short of abandoning all electronics and moving to a northern Canadian deserted mountaintop. Like most Americans, I have nothing to hide, but firmly believe in privacy. Without it, freedom is lost. I've been self-censoring my communications more and more every day, trying to think if what I'm writing will trigger anything at the various agencies monitoring my every thought. This is not freedom. For instance, "trigger" and "freedom" in the same paragraph is probably not a smart thing to do.

Just a list of things in my head that are insecure. Please tell me I'm wrong: Phone calls: intercepted, and access to stored metadata. Emails: intercepted, and access to stored metadata. Texts: intercepted, and access to stored metadata. He goes on. HTTP/HTTPS: broken, and they have a key database. VPNs: broken. They have a key database to use. Cloud: captured in transit, and they likely have access to the datacenter. Computers, files, video, audio, Stuxnet. Smartphones: everything in and out is intercepted, and access to stored metadata. Standalone GPS units might be safe, eh? [Sigh].

Steve: Yeah. I was moved to put both of these in mostly because there's so much of the incoming email reads like this. I mean, and I recognize that the listeners of this podcast, the people who care about security and privacy, who don't just say, oh, whatever, and don't just say, oh, well, it's always been like this, I mean, this is what the listeners of this podcast are thinking about. Certainly we provide, we have been providing, tools for years to begin to work against this. I coined the acronym TNO for Trust No One. And in fact, in talking about Google's ridiculous encryption of their Google Drive, which they can decrypt, we also recently coined ZVE, Zero Value Encryption, as an acronym.

So again, this is still what people want to talk about. We'll talk about it. Soon we'll be

talking about the improvements, essentially coming back from where we are using new technological solutions to begin to restore privacy. I don't think what will ever change, though, is that everything he itemized there is a consequence of the connectivity that we have. It is a huge productivity boost. Anything I want to know, I can Google and know. I mean, that's just - that's amazing. But with it comes some compromise to the fact that somebody monitoring what I'm doing is able to see what I'm interested in. And there certainly is a chilling factor.

I've often talked to Leo about how I'm self-censoring a little bit. Like in Google searches I think, oh, you know, whose attention is this going to come to? Just the other day I was curious about, what was it, the gas that was used in Syria. It wasn't ricin, it was sarin. And so I remembered, though, that Wikipedia was going to be switching over to HTTPS. So rather than just Googling "sarin," which would obviously maybe be seen, I established a secure connection to Wikipedia first, and then I searched for sarin in Wikipedia, that is, with a secure connection, trusting Wikipedia more than I trust Google to keep my interests safe.

I mean, I have no interest other than just I was curious what it was, what was that that sarin gas did? Unfortunately, I found out. But that's the consequence of this sense of being watched at all the time. And this next question by Robert Sutton is just - or actually statement. He talks about privacy in a way I think is really interesting.

TOM: Yeah, he's got a bit about the philosophy of privacy. He's in Brigantine, New Jersey: I can tell that Steve is interested in the philosophy behind the importance of privacy, so I figured I would share a bit of the philosophical basis I use to explain why privacy is necessary: The 20th-century existential philosopher Jean-Paul Sartre asserted that privacy was necessary to make the most out of our lives. This is apparent in his play "No Exit." In this play, a group of people have their eyelids removed and are trapped in a room together. It turns out this room is hell. This is where the quote "Hell is other people" comes from, he says. Their eyelids being removed is so that they can't even close their eyes and imagine that they are alone.

In Sartre's version of existentialism, he claims that humans have two modes of being: being-for-itself and being-for-others. Imagine you're alone in the woods or going for a stroll in the park with no one else around. You look at all the trees, the park benches, and leaves on the ground, and just enjoy the nice scenery. Since you're alone, you almost get the feeling that all these things are there just for you. You perceive these things as objects in your universe. This is being-for-itself.

Then suddenly, you notice someone else in the distance walking towards you, though they don't see you yet. Seeing someone else and realizing they are about to approach you, you now perceive yourself as an object in someone else's universe. So what do you do? You suddenly become conscious of your appearance. You make sure your shirt is buttoned, you fix your hair, you straighten your posture so you look presentable and mentally prepare yourself for an interaction with another person. Then, when the person finally approaches you, you put on a smile, claim you're happy to see the person, and extend your hand for a handshake. In this mode of being, you are viewing yourself as an object in someone else's universe. You're not just behaving as your true self, but you're also behaving how you believe the other person expects you to behave. Viewing yourself as an object in someone else's universe is being-for-others.

Sartre believes that being-for-itself is the mode where humans can be the highest form of themselves and make the most out of their lives. Being-for-others is the source of all shame, embarrassment, and guilt. People who live through the expectations of others and always behave how they believe others want them to behave is what Sartre refers to

as being in bad faith. The philosophy is somewhat derived from Friedrich Nietzsche's concept of "bermensch," which is German for "Superman." An bermensch is someone who is the highest form of themselves with minimal influence from society. An bermensch is always living inside their own head, and they don't view themselves through the eyes of others. An bermensch also doesn't follow any rules or social norms that they don't understand, and make the most of their existence before they croak.

Ever since the whole NSA surveillance fiasco, I realize I'm always considering how I appear to others. Whenever I am talking with a friend, I have trouble getting the feeling of flow, in which I feel like I can be my true self, because I always know someone else is watching. It's like I'm always viewing myself through the eyes of a third party. Before every sentence I speak or write, I enter being-for-others in which I consider how I would appear to someone else listening in on the conversation. It's like I'm always behaving how the government would expect me to behave as the perfect citizen. I even find myself afraid to say inside jokes I have with my friends because I'm afraid someone listening in would take them out of context. I find myself constantly restrained from being my true self.

When the whole PRISM thing got leaked, I could hear Sartre and Nietzsche rolling in their graves. This news made me realize that existentialism is now more relevant than ever. If we believe we are always being watched, we will lose our ability to maintain being-for-itself, and we'll all be living through being-for-others in which we spend our short lives as robots behaving through other people's expectations. This is why in the opening chapter of "1984," Winston Smith sat in the corner of the room as he wrote in his journal, outside of the view of the cameras. He needed to escape the view of others in a desperate attempt to retain his humanity.

I know this is long, but I just thought you would be interested. I'm a computer security grad student, and I started listening to your podcast just to stay informed in current security affairs. But now I find myself looking forward to every Wednesday at 2:00 p.m. Thanks for the great podcast.

Steve: I thought that was interesting. An interesting case.

TOM: You know, I was a philosophy minor, big fan of Sartre, and loved that play "No Exit." There is an alternate take on that. And this doesn't discount anything Rob said. Rob did a great explanation of one view on this. But the alternate take could be that what Sartre was trying to encourage you to do was to be that being for yourself, even in the view of others. I supposed you could take that...

Steve: Ah, be strong enough.

TOM: ...yeah, take that to the logical extent, that something like this actually strengthens us because it forces us to confront that being-for-others and try to resist it.

Steve: It's like gazing around and seeing cameras pointed at you and saying, eh, okay, so what?

TOM: Yeah, exactly. And not feeling like, oh, I have to, you know, some other people are around, so I have to behave differently than who I am. It's being true to yourself. I think we'd all like to choose to put ourselves in that situation, rather than to be thrust into that situation. But the reason this is so applicable, I believe, is Sartre wrote a lot of this stuff based on his experience in France during the German occupation, where he went through exactly the worst form. I mean, we think we've got it bad. They had a much worse form

of a surveillance state.

Steve: Yeah.

TOM: Just without the technology.

Steve: Yeah.

TOM: Shall we go to Paul Durham in Port Elizabeth, South Africa, suggesting an SMTP protocol improvement? He says: Hi, Steve. On Security Now! you have discussed privacy issues relating to email. I have an idea that I think could be applied to SMTP that could improve privacy. Currently when a user sends an email, the email headers show who it is from, who it is going to, and the content of the email is typically visible as well. The mail content can be encrypted by applying PGP. Privacy is still leaked by the header containing the "from" and "to" details.

Suppose each email domain had a PGP key pair. When the email is created by the sender, the recipient is defined in the email as normal, and the email content is encrypted with the recipient's PGP key, as normal. In order to protect the recipient's email address from being viewed in the metadata, the email and the recipient's details should be encrypted with the receiving domain's PGP key. This way the email can get as far as the receiving domain's SMTP servers without the recipient being identified, and only then would the recipient become identifiable to the receiving domain in order to deliver the email to the recipient. Since the domain decrypting the recipient information is the same domain the recipient is in, there would be little, if any, risk of loss of information.

In order to protect the sender's information, the email created by the sender, after it has been encrypted by the recipient's PGP key and thereafter by the recipient's domain's PGP key, can be encrypted by the sender's domain's PGP key. This would protect the email sender's metadata from being intercepted until it is received by the sender's mail server for further delivery. Since the server's domain is the same as the sender's domain, there is little, if any, loss of information at this point.

So to summarize, encrypt an email with the recipient's PGP key, then the receiving domain's PGP key, then the sender's domain's PGP key. This would seem to protect the privacy of an email from the source to the destination, with the exception of the sender's domain knowing the sender, and the recipient's domain knowing the recipient. Does this make sense, and would it work as Paul envisages it?

Steve: Well, first of all, yes. And I got a kick out of this as I was reading it because essentially what Paul has done is create a mini onion router. This is essentially what the onion router does. When you want to send something anonymously out on the Internet, you choose a series of nodes, and then you encrypt from the farthest away one sequentially towards you, successively encrypting, creating these so-called layers of the onion. And then you send this to the first node that's only able to decrypt the outer layer. It sends it to the next node that, because the packet is moving in the same direction that you added these layers, it's able to hop back through, each node decrypting what is then the outermost layer when it receives it.

That's exactly what Paul has suggested here with SMTP. The idea would be that you first encrypt with the recipient's email, then you encrypt with the recipient's domain, then you encrypt with your domain. So then you hand this, that's got three layers of encryption wrapping, to your domain. It's able to decrypt it to find out where it goes. It then sends

it to where it's going, the recipient's domain. That domain can decrypt that layer. Now it knows who it's bound for. It then transfers it to that person. And then that person takes the last layer, the innermost layer off, by decrypting that. So essentially it's onion routing. And we know that the concept is strong and viable, and this essentially applies that to email.

Now, of course, going from the theory, which is sound, to practice is a problem because we all have a protocol which doesn't use much encryption right now. I think that probably email is as ripe, when I was talking about the pressure the notion of surveillance puts on the adoption of protocols, email is probably among the most ripe for this kind of upgrade, switching in general to more security. Initially the optional servers will then be replaced that understand some sort of email privacy, and then we'll begin to get it over time. Because right now it's just almost all of email is completely unencrypted in the clear.

TOM: Do you think email's worth saving?

Steve: That's a good question.

TOM: Or will we just move on to something entirely different?

Steve: It's a good question. I love the store-and-forward nature of it. I love that it's asynchronous. We can do things in the middle of the night. We can answer things when we choose to. Like switching everything to real-time would make us all neurotic, I think.

TOM: Yeah, I suppose that's true. Speaking of neurotic, when I was growing up we all talked about neurotic people going to the mental hospital in Alton, Illinois. But that has nothing to do with Steve, who I imagine is perfectly sane, and from Alton, Illinois. A bunch of my family is over there, too, Steve. Don't worry about it. On this topic of Steve's coffee recipe: Steve, thanks for the great security show. Listener for years. I've heard you talk about this amazing coffee recipe you have. Will you please open source the coffee recipe so the rest of us can get addicted, as well. I'm with Steve. What is this coffee recipe?

Steve: And go ahead and do No. 8 at the same time.

TOM: Okay, yeah. John Hughan also asks about your magical coffee. He says: When you're done working on the far more important webpage that describes the solution to user and web authentication, assuming this isn't something you want to keep as your secret recipe, I'm wondering if you could post your "Guide to 'This is Coffee?' Coffee," including raw materials, hardware, and methods. I have to say I'm very intrigued by your description because I tend to drink coffee only in more elaborate drinks because I've always found black coffee too bitter. But I also hate the number of calories those types of drinks have. So if there is in fact a way to get great-tasting black coffee, I'm all ears, as I would suspect many of our listeners will be, as well. Thanks for a great show.

Steve: Well, I don't want to keep everyone in suspense. I have promised Leo that the next time I come up to Petaluma I'm going to specifically arrange to make him a cup of coffee using my formula so that we have another person's opinion. Because so far, as I have said before, this is the coffee I drink. I mean, I've got it right here, and I've been drinking it. And when I've shared it with people, I've said, here, like, taste this, they're like, is this coffee? They can't even believe how smooth and perfect it is.

But so here's the deal. I mean, it's not a big mystery. I did search for it for a while. The way I got to here was I was making what Starbucks calls Americanas, or Americanos, at

home. I have a commercial espresso machine. And so I would nuke a large mug of water, making it hot water, and then drop the espresso directly into the hot water. I'm big on not having the espresso go into a little shot glass because it immediately starts to oxidize and goes from, like, light brown to black and then becomes very bitter. I don't know how anybody would want to just drink espresso that's been sitting around in a cup for a while. But by having it drop immediately from the filter holder into the coffee, it gets diluted, and it's protected from the oxygen.

And so I was making a very nice cup of coffee. The problem was it only makes one cup. And so then you're going back, if you want to have five cups over the course of the day, you're going back there all the time. So what I ended up evolving to was to use Starbucks espresso bean and drip brew that. So my formula is pretty simple. I have a burr grinder, and I've got the one that I told Leo about. I don't remember the model number now. But I chose it because it has no reservoir. The beans drop right through it down into the coffee filter.

So I take, I think it's one ounce, I don't remember the amount, but I have a measuring deal, essentially for about five cups of coffee. And so I take the beans, grind them through this, so they grind and drop right into the filter, and then it's just do a drip brew, a sort of a small, sort of like a half pot. It's a - and it's also the drip brewer that I've talked to Leo about, the little simple coffee percolator. And it uses the little Melitta disposable, I use the brown filters because I don't want the bleach, and drops right through into the pot. And then I transfer it to something that keeps it hot for hours at a time. And that's it. So anyone who's interested, you need a good grinder. You can't use one of those little spinning blade things that just fractures the beans.

TOM: It hurts me just thinking about that, yeah. Don't use that.

Steve: I know, it's horrible. So you need that so you get a consistent grind. I've got mine set to 3, whatever that means, which is sort of large, I think, because you don't also want it to be ground too fine, or you get a more bitter result. So anyway, I will do all this the next time I'm with Leo, hand him a cup, and see what he thinks. And in the meantime, anyone who wants to experiment, just buy a bag, you can get a one-pound bag of espresso bean, you don't want it pre-ground, espresso bean. Unless you don't have a grinder, in which case they could grind it, then you could run home and give it a try and see what you think.

TOM: Sprint home, yeah.

Steve: Yeah.

TOM: I like this idea of the grinder that goes right into the filter. Because I have a burr grinder, as well, and I've been meaning to replace it because it's old, and it's also really loud. And it goes into this plastic thing that you dump the coffee out, it's not good. So I love that. If you remember what brand that is, I'd be curious.

Steve: I will.

TOM: Simon Comeau-Martel in Montreal, Quebec, Canada wonders about Touch ID on the new iPhone 5S. There were definitely people earlier in the chatroom wanting to know this, too: What's your take on the new Touch ID technology by Apple, Steve? What's its impact on the device disk encryption? Historically, the PIN/password was used to protect the encryption key. How can we fit a fingerprint reader in the equation? It will never scan exactly the same thing twice. Also, should we worry about someone lifting prints from a

desk, or from the phone itself, to try unlocking it?

Steve: Okay. So at this point all we have is conjecture. I'm looking for and have not yet found, so I'd like to enlist the entire listenership of the podcast as a dragnet for information. I'm really curious to know exactly what Apple has done. Simon is right that certainly every time we put our finger on the scanner, it's going to be a different image. We seem to know that it uses detail extraction. So it's not actually using the image. It's using the well-known fingerprint details, loops and whorls and breaks in the ridges of fingerprints which are uniform over time.

So the idea is that you train it by putting your finger on the scanner many times, giving it the same fingerprint over and over and over. It so-called "learns" that. So what it's doing is it's not memorizing the image. It's memorizing the features. And then it decides that your fingerprint has the following features in a certain topology. That's what it memorizes, and then that's what it no doubt uses to match with in the future in order to unlock your disk encryption.

So it can't, I would be very skeptical if it used that as the key directly because, for example, you would never be able to retrain it with a different one. There's got to be an intermediate step where it says, is this the same one I've been taught to accept? And if the answer is yes, then it uses that. And maybe it turns those features into a hash, and then it uses the hash. We don't have enough details yet. But it certainly is the case that it goes through some iteration of, like, feature recognition. I'm sorry, go ahead.

TOM: No, no. Am I right that - I thought I remembered them saying that, even though you have the fingerprint, you still set a normal password as sort of a backup...

Steve: Yes.

TOM: ...to the fingerprint not working. So could that be what it's relying on for the consistent behavior, and just working the way it's always worked?

Steve: Well, yeah. So the idea would be it has a template for what it has seen. I was glad to see that from day one Apple recognized the privacy concerns. So, and I hope they will, like, put all of our minds at ease by just telling us what they've done. They probably have some patent stuff they may have to wait for. But we need to know. But, for example, if you had a means of representing the topology of the features, so that loops and whorls and gaps are mapped, and that's invariant over time, then you could hash that to create a hash key that was invariant over time. So at this point we just sort of have to make things up and wait till we hear exactly what they've done. I love that they've done this because, I mean, I watched Leo and Sarah reacting to the news as it was happening, and Sarah talked about how she hates having to enter her long, complex, secure password every single time she wants to buy something. She just wants to put her finger on the phone, which I do, too.

TOM: Java678 in the chatroom gave us a link to one of the Apple patents...

Steve: Ooh.

TOM: ...that they have used for fingerprint identification. We still don't know if that's what they implemented, though, because it shows things like near-field communication and stuff involved which are not in the iPhone 5S.

Steve: Right.

TOM: We've got one last question here. Macarthur in Virginia shares some thoughts about the randomness of Intel's RNG: Intel claims that it's purely random, but they haven't proven that it isn't "not random." They've only shown that it appears to be random enough. You can never prove that something is random. All you can do is prove that something isn't. So all they've shown is that it doesn't break down and lose to the tests that we currently have. Thus, they've shown that it seems to be random. They've shown that it is enough random that it doesn't show bias and seems to be random. Any RNG can't be proven to be random. It can only be proven to not be random. Now, while I do think that Intel's fab is probably the best in the world, they can't verify every single one. That's the words of our writer, Macarthur here. They can't prove every single one. That's why the Linux kernel doesn't rely solely on Intel's RNG. They mix in Intel's RNG products with its own internal pool of random data.

The kernel gets it from network traffic latency, all of them; HDD access times; temperature from the various parts and a few other places. So they currently mix that data from Intel's RNG with that data to make it stronger and more random. So even if Intel put a backdoor in it, it'd never result in completely unrandom data since the kernel stores that data in RAM. And the CPU could go through and try to turn the data to all zeroes, but the kernel would likely reject that result as it tries to keep that data as random as possible.

The last I remember reading, the kernel held about 4 to 16Kb of random data on the /dev/random device. I believe that the BSD kernels use a similar approach. The Linux kernel, and as far as I know the various BSD kernels, don't just accept what intel feeds them. They use it as an additional source of random data. So all Intel is doing is making that data more random; and, at worst, they're keeping it as strong as it was before.

Steve: We've talked a lot about randomness because it's crucial for cryptography. And one of the - in fact, we highlighted it last week when we were looking at the accusations that the NSA, through the NIST, may have deliberately subverted an almost never used, and no one really cares about it anyway, random number generator. But the idea that you might have something nonrandom is a huge concern.

I love this notion of mixing random sources. I've used it myself. Listeners with a really good memory will remember when I was messing with the Off The Grid paper-based encryption system [SN-315]. I needed an ultrahigh entropy pseudorandom number generator. Not that 256 bits wasn't enough, except that there are so many possible Latin squares of that size, I mean, just a phenomenal number of possible ones, that if I used a random number generator with a small amount of entropy, it couldn't have that many states, and it couldn't give me nearly as many Latin squares as possible.

Anyway, the point is that I have GRC provide entropy, whatever it was, 256 or 512 bits, and I have JavaScript in the user's machine provide it, and mix them together. It's one of the coolest things about randomness is that - and it's sort of what we see with the XOR feature. It's sort of one way to think of it is you can't unscramble an egg. Once you've got something random with a lot of entropy, there's no way, nothing you can do to it can lower the level of entropy it has. It's got it. And all you can do as you add more is increase the level that it already has. Which is why the idea that the kernels would take sources that they already maintain, things like packet arrival times, hard disk completion events, use high-resolution timers to watch things happen. And that's going to, I mean, technically they aren't absolutely random. But what they are is absolutely unknowable to an attacker.

So you take all those different sources. And by all means, if the Intel chip wants to throw

some bits at you, take it in. It can't hurt you in any way,. Even if it sent all zeroes, you'd still have all the other random sources that are pooling their randomness. And this whole notion of, when he talks about the kernel having 4 to 16K, that talks about the total entropy in a so-called "entropy pool." And as you pull randomness out of the pool, it diminishes the pool size. And then, as other events happen in real time, it begins to replenish the pool size.

So it's funny how far we've come in our understanding of what a good random number generator is. Back not that long ago, really, we were using a simple equation, like a linear, what's called a linear congruential pseudorandom number generator, which just was an addition and a multiplication and produced an absolutely deterministic series of numbers. I mean, just an awful source of random numbers. But that's what people used. Today, we've gone from much more sophisticated pseudorandom numbers to this notion of a pool of entropy that has a known size that we are pulling from and adding to as the entropy is being needed and being replenished. So anyway, just the whole notion of entropy is key to cryptography because ultimately we are protecting secrets, and it's a random number that is the secret that we don't know.

TOM: Yeah, and as somebody pointed out in the chatroom, it takes something like a black hole if you want to unscramble an egg. So that's pretty good security. For now, anyway. For the foreseeable future. I want to go take a dip in the pool of entropy now. Be more safe.

Steve: I wonder what color it is. Probably is.

TOM: Yeah. I wonder what kind of swimsuit I should wear.

Steve: Maybe it's 17% gray. It probably is.

TOM: It probably changes. I guess. Just fluctuates. Well, Steve, that brings us to the end of a great episode of Security Now!. I'm so glad I get to do a couple more of these with you.

Steve: It's going to be really fun, yeah.

TOM: Yeah, I will be here for two more weeks. Leo is on vacation. He'll be back in a couple of weeks. Meanwhile, folks, if you have not, I can't imagine you haven't, but if you have not gone to GRC.com and checked out all of the amazing things, you were talking about the Haystacks thing that you were doing, you were talking about the password thing, the new version of SpinRite, ShieldsUP! still going on there, you've got to go check that out, folks. Any last thing to mention before we head out of here?

Steve: Only that I do keep asking people to send your thoughts and questions to GRC.com/feedback. And also I will say again, what we don't have is a beta of SpinRite 6.1. What we do have is a very active group that's been playing with the code that I've been writing over the last couple months. I get email from people saying how do I join that? GRC.com/discussions will explain how you participate in our forums. We don't have web-based forums. We're old school NNTP newsgroups. The whole population of people really prefer that. I like it. It just seems, I don't know, more hardcore, and it's more about being down to business. Also very high quality posts and people hang out there.

So you'll have to go to GRC.com/discussions in order to understand how to hook up to our NNTP server. You can't go to news.grc.com with a web browser. There's no web server there. You have to have - I think Thunderbird has one. I use Gravity on Windows.

I know that there's NNTP news readers for Apple. I use something called NewsTap on my iOS devices, on my various iPads and iPhone. I like it a lot. So anyway, we'd love to have people show up. The more testers we have, the merrier. And it's a lot of fun to participate in nailing down this code.

TOM: Absolutely. Go check it out, folks. And don't forget about our show notes, too, at TWiT.tv/sn. We'll see you next time.