SQRL: Anti-Phishing & Revocation

Description: After following-up on a week chockful of interesting security news, Steve and Leo continue with their discussion of SQRL, the Secure QR code Login system, to discuss two recent innovations in the system that bring additional valuable features.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-426.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-426-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson has the latest - yes, another Java patch - and more information about Lavabit. More information about the NSA, too. What is a Ferret's Cannon? It's coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 426, recorded October 16th, 2013: SQRL, Anti-Phishing, and Revocation.

It's time for Security Now!, the show that covers your privacy and security online. Somebody in the chatroom before we began the show today, Steve Gibson, Explainer in Chief, said we should call it "Insecurity Now!" since it really talks mostly about insecure. But not today. Today we're going to talk about better security.

Steve Gibson: Well, yeah. Actually, this is one of those episodes where so much happened in the last week in security news that, I mean, we just - there's a whole bunch of really interesting stuff to talk about. And in the past when we've done that we've just said, okay, we're not going to have any major topic because too much happened. Yet there was also some news over in SQRL land, SQRL, the Secure QR code Login system, where we made some advances in antiphishing protection, which is something that people have been concerned about. And also the issue of revocation, that is, revoking your credentials on a site, because those are both things that are - for example, because of the simplicity of the system, these things are more difficult.

For example, when you have a public key infrastructure, you can revoke a certificate. In fact, one of our stories today is how GoDaddy revoked Lavabit's certificate once the FBI had it. So thanks to the involvement of a third party, you can certainly achieve revocation. But how do you do that in a two-party system? And of course one of the big huge benefits of SQRL is it is deliberately two-party. It is Trust No One else. Would that be TNOE? Anyway, we don't want to add an "E."

Leo: 2NO.

Steve: Just TNO. But anyway, huge, a ton of really interesting news for the week. And then I want to talk about some, basically some advances in the way SQRL operates, which - and my god, Leo, we're just seeing an explosion of interest, huge number of projects started. There's even a presentation at next week's HTML5 Developers Conference at Moscone Center in San Francisco on SQRL.

Leo: Oh, that's neat.

Steve: Yeah.

Leo: Who's doing that? Not you.

Steve: It's, no, I'm not doing it. No, somebody - and then there's another, there's an annual identity conference, and it's going to be presented there, also.

Leo: Wow.

Steve: So, I mean, it's, yeah, it's taking off.

Leo: Very exciting. Well, good. All right. Well, so I guess we'll start with security news. And it wouldn't be security news if there weren't a Java update.

Steve: Oh, and baby, do we have one today. This is - you want to be at Java 7, Update 45. That's the newest one. Fifty-one security vulnerabilities, all but one of which are remotely exploitable without any authentication.

Leo: Oh, that's not good.

Steve: So 50 out of 51 are, if the bad guys know this, you're hosed, basically.

Leo: Wow.

Steve: So it's just phenomenal. And but the problem is, it is a full-feature big language that should have never been put online. It's not a problem, as we've said often, if you have it on your desktop. It's when Sun, and then of course Oracle continuing this, when they decided that it would be a good idea to make this a browser plugin - oh, look, just think what we can do if you can download Java applets. You could have full Java features on your browser.

Well, listeners to this podcast know that's just not going to turn out well. I mean, Flash is sort of a - is a subset of that. JavaScript is a subset of even Flash. And all of them have problems. It's just - it is a bad idea to allow a website to run code on your computer. That's never going to turn out well. So, I mean, because it's so valuable, I mean, no one's going to argue that it really makes the web come alive. I mean, it's Google's entire world is based on now running apps from them on our computer. That's how Google Docs works, and all of - many of these cloud-based systems are working that way. So we're not going to succeed in absolutely saying no.

But it's the reason, for example, that I wouldn't consider operating without NoScript in Firefox, is that my default is no, just as it is for every firewall now in existence. Firewalls used to be, allow it in unless we know we want to block something. Now all firewalls are block it unless we know we want to permit something. And that's what NoScript does. And when you configure it properly, it's not a problem. Some sites don't work until you say, okay, yeah, I'll allow you to work this time, and then they come alive. Of course, HealthCare.gov never works, whether you enable it or not.

Leo: And it's in WordPress.

Steve: It's 60-some JavaScript files. Talk about a JavaScript nightmare. I don't even think you could permit it to work in NoScript. NoScript would probably crash if you tried to go in and individually enable all the JavaScript that the HealthCare.gov website needed in order to function.

Leo: No, I really want to emphasize, and I know our audience knows this, Java is not JavaScript. They're two different things which are inappropriately similarly named. You know, I was really peeved, just before I left, actually I guess it was last week when I got back, I needed to contact Comcast. We're moving, and so I wanted to move. And the only way you can move with Comcast is using their Java-based chat client. So I had to - normally I don't have Java installed on my machines.

Steve: No kidding.

Leo: So I had to install Java.

Steve: So it's like, all other methods of contact are now, sorry, this is...

Leo: Well, they don't, you know, I guess you could call them. But they really don't want you to. And you can't do it just on the web. You have to go through this person. So I was just looking, and you reminded me, oh, god, I've got to turn off - because I had to allow it, and now I have to un-allow it again so that I'm not - it's just terrible. Terrible.

Steve: Yeah. So I will reiterate that, first of all, you will, by this point, you will know if you are a person who needs Java. Whenever we talk about it, I get tweets from the Scandinavian countries that say, well, all the banks over here require us to have Java. The good news is I've had some updated tweets that say they're moving away from that.

It's like, okay, good. So this is something we just need to kill because here we are at Java 7 Update 45. We've talked about how Java 6 no longer has updates, yet, okay, here's 51 security vulnerabilities that also exist in Java 6, that will never be patched.

Leo: Wow.

Steve: So this is what's happening is the bad guys are seeing the mistakes which are bubbling up and being fixed in the current version. When they look at the patch, they figure out what got changed. That tells them what was fixed. Then they go looking, deliberately looking, it gives them a pointer to the problem in prior versions which are not being fixed. And of course this is the same problem we expect to be seeing when XP goes out of service in 173 days. I have my XP counter on my Windows 7 machine.

So the only strategy, first of all, you probably by now know you need it, if you do, just as you had the experience, Leo, of needing to use it in order for a specific purpose, for a window of time, and then saying no. The best solution, because we've all got browser choices now, is to remove it from your go-to browser of choice, that is, remove the Java plugin so that Java won't run on that browser. And you could verify doing that. I have, actually have a Java applet on GRC, that big number calculator which you can access at GRC.com. It's completely benign, but it's a nice way of verifying, oh, good, it doesn't work, because you don't want it to work unless you absolutely know you do.

And then only allow Java to run in, like, your third-choice browser, your browser of last resort. If you've got no other choice, that's the browser where you want Java to be. That way you're not using it by mistake, and it doesn't have an opportunity because the point is normally the browser will request a Java applet by default. So all you have to do is touch a website which is either malicious or has been hacked to install, like, a malicious ad or a malicious little tag. The browser sees it, goes and grabs that Java code and runs it. That's the default behavior of browsers with Java. So it's - you just can't allow that to be.

Leo: We must not allow it.

Steve: We must not. So put it on a browser you never use, and only if you need to talk to Comcast in order to re- after you move a home.

Leo: So frustrating. So frustrating.

Steve: Yeah. Now, speaking of frustrating, there was a really sad piece of news that is an upshot of the NSA revelations, more blowback from what Edward Snowden showed us. And this is a news alert posted on the InternetGovernance.org site on the 11th of this month. The headline was "The core Internet institutions abandon the U.S. Government." This is not as significant, for example, as if the dollar stopped being the world reserve currency, but it has that feeling. We've sort of - we've had ICANN and IANA, I mean, because the U.S. invented the Internet, we developed the technology, we basically - taxpayer money funded the Advanced Research Projects Agency, ARPA, back in the day, to experiment with the concept of packet switching.

And back in the packet-switching podcasts that we did we talked about how - what an

amazing insight it was that you could achieve the same thing as dialing up your modem, as we used to in the old days to connect to a service, you could achieve the same thing with sending little packets of data that just somehow found their way to their destination, completely a different concept, which people said, "Wait, wait, wait a minute, how do I know it's going to get there?" Well, you don't. "But how do I know when it's going to get there?" Well, you don't. But it's good enough.

And it turns out if you layer, on top of that uncertainty, you layer protocols that provide for order of arrival and guaranteed arrival and sort of make up for the fact that the underlying architecture is weird, sort of nondeterministic, the whole - it actually works, and we're all using it. So what's happened, unfortunately, is that the world has said, uh, we don't think we should leave these organizations with the U.S. anymore. So just the first two paragraphs of this sad announcement said: "In Montevideo, Uruguay this week..."

Leo: Montevideo.

Steve: Montevideo, thank you. Montevideo.

Leo: Montevideo.

Steve: Monty's Video. Montevideo, thank you, Leo, Uruguay, "the Directors of all the major Internet organizations - ICANN," the IETF, another one that we often talk about...

Leo: The Internet Engineering Task Force.

Steve: Yup, "the Internet Architecture Board (IAB), the World Wide Web Consortium," that's the W3C that we often speak of also, "the Internet Society, all five of the regional Internet address registries - turned their back on the U.S." - I know, thank you for the heavy sigh - "turned their back on the U.S. government. With striking unanimity, the organizations that actually develop and administer Internet standards and resources initiated a break with three decades of U.S. dominance of Internet governance."

Leo: I'm not sure that's so bad, really. I mean, in general, for the world, I think it's probably good. Why should we dominate?

Steve: Well, no. But that's just it. There's - I agree with you. This is the right outcome in the same sense that I believe...

Leo: For the wrong reason.

Steve: ...having it known, yes, in the same sense that for the reason I believe having it known what the NSA has been doing is the right outcome. But it's a black eye. Or maybe it's a bruise. I may be - we won't go so far as a black eye. Maybe it's a pimple. So, "A statement released by this group called for 'accelerating the globalization of ICANN and

IANA functions toward an environment in which all stakeholders, including all governments, participate on an equal footing.' That part of the statement constituted an explicit rejection of the U.S. Commerce Department's unilateral oversight of ICANN through the IANA contract. It also indirectly attacks the U.S. unilateral approach to the Affirmation of Commitments" - that's capital "A" and capital "C," that's a formal "pact between the U.S. and ICANN which provides for periodic reviews of its activities by the GAC and other members of the ICANN community." And it says in parens, "(The Affirmation was conceived as an agreement between ICANN and the U.S. exclusively. It would not have been difficult to allow other states to sign on as well.)" But that's not the way it was done. So it's like, okay, well, yeah.

Leo: Yeah, I think this is not inappropriate, though. It's time, it was - it seemed odd that we controlled it as much as we did. I don't blame other nations for not being thrilled about that. So, but what happens? Is it the U.N. now? I mean, what is the - is there a body at all? Or is it just these are all just multigovernmental? Or NGO, nongovernmental?

Steve: Right. They're NGOs. They will, I mean, and they've all got committees and teams. And so they'll just basically cut themselves loose and float off and no longer recognize even a modicum of preferential rights by the U.S., which...

Leo: Yeah, I think that was...

Steve: ...arguably they used to have.

Leo: That's a good idea.

Steve: And I agree with you. I agree with you. But we know why it happened.

Leo: Yeah, yeah, yeah. But, you know, there you go. This precipitated a lot of good. It revealed a lot; you know?

Steve: Yes.

Leo: And this stuff has been floating - it's not like - the stuff Snowden revealed has been floating around for a decade. And this just forced everybody to pay attention to it, I think.

Steve: I wouldn't say that, actually.

Leo: No?

Steve: We got a ton of facts. We learned a lot, Leo.

Leo: We suspected a lot.

Steve: Yes. It was in the air. But it's very different to have slide presentations with them boasting and chuckling about what they're doing. That's more than what we thought. And on that note, Yahoo! on the one-year anniversary, upcoming, not yet, we still have a few months away, it will be on January 8th, Yahoo! has announced that on the one-year anniversary of allowing, that is, supporting HTTPS connections, only four years after Google did, this coming January 8th they will be on by default.

Leo: Yay.

Steve: So the checkbox which is normally off will be on.

Leo: Good.

Steve: Starting January 8th of 2014.

Leo: That's fabulous news.

Steve: So that is, that is just, you know, better late than never, I say. I don't know, I haven't looked at their servers to see whether they're - whether they've implemented perfect forward secrecy. We certainly hope they have because it's now time for everyone to step up and do that. We've got some interesting news about that, too. Oh, and in fact we're stepping into it because, as I mentioned earlier, GoDaddy revoked Lavabit's certificate. Now, this news was initially, like, yay, you know. And, like, I tweeted this news. And a lot of people came back and said, really? They care? Or I'm surprised. Or I give them more credit than I - and I would be surprised if it was their idea. I think it was probably Ladar who said...

Leo: He says please revoke my certificate, yeah.

Steve: Yes. And who knows what the timing was? The news came out. But Leo, you ought to just go, if you want to show this on - just go to https://Lavabit.com. And when I checked this morning, before reminding myself that I was going to mention it, you can see right there, Firefox, my browser du jour, or of choice actually, says, whoops, this certificate has been revoked.

Leo: It says you might - Safari says, "You might be connecting to a website that's pretending to be Lavabit, which could put your confidential information at risk." That's exactly the behavior you want. That's good because in fact it might be an NSA front.

Steve: Right. So anyway...

Leo: What happens if you continue, just out of curiosity? Do you get - oh, you still get there. But you get his note, Ladar's note saying I'm out of business, yeah.

Steve: Correct. Now, what we don't know is when he implemented perfect forward secrecy. But the good news is he has it now. So, and again, we may well have always had it. I was impressed, though. SSLLabs.com has added the ability to check a website's support for perfect forward secrecy. And Ladar Levison's Lavabit.com servers, which as you just saw are still online, even though - and in fact, SSL Labs notes that the certificate has been revoked, so it sees that and flags that. But he did float the Ephemeral Diffie Hellman Key Agreement Ciphers up at the top of his server's recommended and supported ciphers list. So, and those are the ones that - we did a whole podcast not long ago [SN-412] on this whole - on perfect forward secrecy and how it's because traditional ciphers use the certificate, essentially do not have separate keys for authentication and signing, but rather they use the same key, that is, the server's secret key, for both authenticating their certificate and for signing the session. That means that, if somebody in the future were to get the key, they could decrypt past conversation.

Well, using perfect forward secrecy ciphers prevents that from happening because then you are negotiating a so-called ephemeral key on the fly that only exists between those endpoints until it expires. But it means that capturing the server certificate doesn't give you an advantage in being able to decrypt old traffic. So the good news is Lavabit's servers are configured that way. Again, we don't know - it matters when they were configured that way because traffic that may have been captured prior to then, and we now know that the FBI does have the old certificate, so old traffic could be decrypted if perfect forward secrecy had not been in place all along.

And, remember, I was never really impressed with all of this because it's email. And you can't really encrypt email. Paying customers got their data at rest encrypted. That is, it would have come in probably unencrypted. Even if Lavabit supported connection encryption, both ends have to in order for that to be successful. And many web servers still today don't. So it would have been unencrypted on the wire. Then he was encrypting it while it was stored. And then of course when it was going back out it would be unencrypted unless you were connecting to his server securely.

So again, it takes a lot for email to really be encrypted point to point. In fact, it takes PGP, GPG, GNU Privacy Guard and all that extra layer stuff. You really want to encrypt it in your browser or in your email client all the way to the destination email client. That's the way to be sure. So he was offering a service better than not, but we don't really know what it was the FBI got and whether there is an archive of previously encrypted traffic that the old certificate still allows them to decrypt.

But the other news from Ladar is that he has briefly brought Lavabit back up. Leo, if you go to liberty.lavabit.com, that takes you to a new page. And he says, "Due to concerns about the continued integrity of customers' passwords, we are offering a..."

Leo: Hmm. I'm not getting anything. Liberty, oh, did I spell it right? Liberty...

Steve: Liberty.lavabit.com. He says, "Due to concerns about the continued integrity of customers' passwords, we are offering a short window of five days in which users can change their password before we allow anyone to download an archive of their stored emails. The download functionality will be available starting this coming Friday, October

18th...

Leo: Oh, that's why. It's not up yet.

Steve: I brought it up. Came right up for me, https:// - oh, did you do a secure link, https?

Leo: No, I didn't. I just typed "liberty." So let me try again.

Steve: Ah, liberty.lavabit.com. So starting this coming Friday, October 18th...

Leo: Yeah, now I've got it, okay.

Steve: Okay. At 7:00 p.m. Central time. He says, "Since the SSL certificates formerly used to protect access to Lavabit have been compromised" - notice you're using a new GoDaddy certificate, https://liberty.lavabit.com. "Since the SSL certificates formerly used to protect access to Lavabit have been compromised, we recommend manually validating" - and this is unfortunate; but, okay, I'll explain why in a second, "the serial number and fingerprint your computer received before using this website." So then he shows them.

Leo: Yeah. So that would be pretty easy to spoof that, then, is what you're saying.

Steve: Yeah, that's the problem, is if it's compromised, they're just going to put this - they're going to change the page to show the serial number and fingerprint of their spoofed certificate. So what you need to do is, first of all, the best thing to do is to ask for details, certificate details, and look at the certificate chain. You will see liberty.lavabit.com at the end of the chain. Then you will see, one step up, GoDaddy secure certification authority. And then the root should be, and this is in Firefox. The different browsers may have different roots, but in Firefox it says "Builtin Object Token: Go Daddy Class 2 CA." So that's the chain. If there's anything different than that, then something is intercepting your SSL connection. Unfortunately, it doesn't work to show the serial number and fingerprint in the page you're trying to verify because, as I said, if it's been compromised, that'll be changed to the compromised certificate. But, again, his heart is in a good place.

Speaking of which, he wants - he's got a money-raising operation. If you go to, again, https://rally.org, r-a-l-l-y dot org, that takes you to the home page of a money-raising site. And his page is rally.org/lavabit. His goal was $96,000, and he's just about there. He's at $93,876 when I looked this morning. So that is Ladar hoping to raise money for his legal defense fund in order to deal with the fact that he's basically fighting the government over what they're doing. And he has said that he hopes, maybe, if things work out, he could bring Lavabit back in a way that he's comfortable with in the United States. He's not willing to move it offshore. That's - he lives in Texas, and he wants to stay in Texas. So there is that site, rally.org/lavabit, if you want to learn more about it. He's got a neat little video where he spends most of his time scratching his dog.

Leo: I see the dog. He's a nice man. He has a dog. He's a dog lover.

Steve: He is. He scratches - his dog jumps up on his lap in the middle of the video.

Leo: Well, how could he be a bad man if he has a dog who loves him?

Steve: I think it's very clear his heart is in the right place.

Leo: Yes, yes.

Steve: No doubt about it. So, and of course in the news another day, another data collection program revealed of the NSA. This was in the news this week. BoingBoing covered a report that the Washington Post had. You may want to bring the Washington Post slides up, Leo, that's that second link in my show notes, where it is revealed in some additional documents that the NSA has been collecting half a million, okay, 500,000 buddy lists and inboxes per day.

Leo: You can have my buddy list. Really? Buddy lists?

Steve: Buddy lists, yes.

Leo: Terrorists have other terrorists on their buddy lists?

Steve: From instant messaging clients. Well, they want to build - the whole deal is networking. That's what this metadata, email metadata gave them is that, even if they didn't have the content, they had - they were able to build networks of inner connectivity. And it's clear that that has a tremendous value to intelligence gathering. So the Washington Post said: "Rather than targeting individual users, the NSA is gathering contact lists in large numbers that amount to a sizable fraction of the world's email and instant messaging accounts. Analysis of that data enables the agency to search for hidden connections and map relationships within a much smaller universe of foreign intelligence targets. During a single day last year" - this is from the documents and the slides that the Washington Post showed - "the NSA's Special Source Operations branch" - okay, one day - "collected 444,743 email address books from Yahoo, 105,068 from Hotmail, 82,857 from Facebook, 33,697 from Gmail, and 22,881 from other unspecified providers."

Leo: Now, I'm looking at this. Is this done by malware or with cooperation of these providers?

Steve: This is the tap that I talked about in my first surmise of...

Leo: Upstream.

Steve: Yes. They are passive taps provided by Internet service providers that allow the NSA to monitor all this traffic.

Leo: Not Yahoo!, not Google, but in fact their Internet service providers, their upstream providers.

Steve: Like AT&T that's a major, yeah, Tier 1 provider. So this says: "...according to an internal NSA PowerPoint presentation. Those figures, described as a 'typical daily intake' in the document, correspond to a rate of more than 250 million per year." Okay, so they can multiply, half a million a day. "Each day, the presentation said, the NSA collects contacts from an estimated 500,000 buddy lists on live chat services, as well as from the 'inbox' displays of web-based email accounts." So that's interesting. The web-based email account, when the web page shows you your inbox, they're getting all of that, basically all of your inbox contacts.

Leo: Hmm. Hmm.

Steve: Yeah. So...

Leo: That's interesting. So I think that this points to the fact that the metadata is really, in many cases, more valuable than the content, is who you talk to, who they're talking to, building a web of communications.

Steve: Right.

Leo: And then you have a person of interest, you can see who they talk to and who their friends talk to.

Steve: Right. The way to think about this is where, if you consider our podcast, a group of smart people, we're a bunch of techies that understand this stuff. We know how the Internet works, how this technology works. We know what can be done. So consider then the NSA is a well-funded, financed, organization of people every bit as smart as the people who invented and designed the Internet, and they're doing everything they can. That is, what are they doing? Everything possible.

Leo: Everything they can.

Steve: Everything they can. If any of us smart people could think of something that could be done, the NSA has some wacky names for it, and it's on some slides, which we either have seen or will be seeing. And in fact we have an interesting article from Bruce Schneier, who's got some eye-opening names of NSA programs.

First I wanted to mention that there was a somewhat sad commentary about whether the Do Not Track initiative might be dying. This was also in the Washington Post. The headline was ""The Internet's best hope for a Do Not Track standard is falling apart. Here's why." It goes on to explain. And basically it's what we would expect. I'm not hugely disheartened because all I wanted was the technology to be there. Just get the header into our browsers. And everyone thought, well, okay, but, Gibson, that doesn't mean anything if people don't support it. And it's like, I know. I understand that. But the wheels of justice turn slowly. Legislative things happen slowly. Once we have that in our browsers, then there's a chance for laws. Until we have that in our browsers, there's no chance.

So my feeling is this has all been worthwhile. It certainly didn't cost us anything. There wasn't any effort on our part. There was a lot of effort that went into people really trying to make this happen. In the subhead on the Washington Post, they say, "Should businesses be forced to stop tracking your movements on the Internet?" And I'll share just a few paragraphs of this because it wound up with the EFF making their position known.

The Washington Post writes: "It sounds like a simple question. But judging by the growing despair among members of a diverse group assigned by a standards body to resolve just this issue, the answer is hardly clear. The task force itself is deeply divided. In a member survey completed Wednesday" - that's last Wednesday - "half of respondents, albeit a minority of the entire working group, said the negotiations weren't working and should be abandoned. 'This proceeding is so flawed, it's a farce,' wrote Jeffrey Chester, executive director of a privacy group involved in the talks, in a comment. 'Global online users deserve better.'" So he's just disgusted.

"The working group is affiliated with the World Wide Web Consortium (W3C), the official custodian of web standards. It was initially brought together to develop a negotiated approach to online behavioral tracking. The collection of ad companies, privacy advocates, and outside experts" - I mean, you couldn't ask for oil and water to be mixed together - "were supposed to settle a longstanding debate about consumer privacy and help determine the future of advertising technology. But what began as cautious engagement among these [admittedly diverse] groups has devolved into open revolt against the process."

And so here's the EFF saying, quote, "'We appreciate the efforts of the W3C and all the chairs to date,' wrote Lee Tien, a top lawyer with the Electronic Frontier Foundation. 'But EFF has lost confidence,'" he writes, "'that the process will produce a standard that we would support. We therefore prefer that the group simply end. If the group continues, we [the EFF] would seriously consider dropping out.'"

Leo: Wow.

Steve: Yes. "The impending collapse of the Do Not Track conversation is part of a broader failure to agree on what obligations advertising companies have with regard to online tracking, and what the word 'tracking' even means." So anyway, I mean, no one expected this to be easy. I mean, and in fact no one expected it probably to even be possible. But this was the sort of thing, well, you want to try it and hope that maybe something could be resolved, and it doesn't look like that's the case.

Leo: No, no.

Steve: Meanwhile...

Leo: Meanwhile, back at the ranch...

Steve: Yes. Well, there's some interesting stuff here. Bruce Schneier, our famous cryptographer who's become very involved in this and almost, I wouldn't quite say an activist, but certainly a spokesperson for the Internet freedom and privacy side, who understands the technology, wrote an interesting piece in the Atlantic which I want to share because it's some additional news with lots of specifics about - which I would reduce to saying that what the NSA is doing it turns out is not even passive, or is not only passive. Everything we've been talking about so far has been passive eavesdropping on the part of the NSA. There's a program called FOXACID which is, again, their name, which is anything but. And so this is Bruce, remember, not some random alarmist. And he couched this description in wanting to help us understand how the NSA thinks about security and risk. And he said: "At this point, the [agency] has to assume that all of its operations will become public, probably sooner than it would like." And he said - and so this was, this what I'm going to share was written in the Atlantic, but it was also posted in the Guardian.

He says: "As I report in the Guardian today, the NSA has secret servers on the Internet that hack into other computers, codenamed FOXACID. These servers provide..."

Leo: I love, by the way, all of these names like Ferret Cannon and FOXACID. Somebody's got some sense of humor, anyway.

Steve: I know. Oh, goodness. FOXACID. "These servers provide an excellent demonstration of how the NSA approaches risk management and exposes flaws in how the agency thinks about the security of its own programs. Here are the FOXACID basics: By the time the NSA tricks a target into visiting one of those servers, it already knows exactly who the target is, who wants him eavesdropped on, and the expected value of the data it hopes to receive. Based on that information, the server can automatically decide what exploit to serve the target, taking into account the risks associated with attacking the target, as well as the benefits of a successful attack." So it's a complete cost benefit analysis of actively exploiting a target.

"According to a top-secret operational procedures manual provided by Edward Snowden, an exploit named Validator might be the default, but the NSA has a variety of options. The documentation mentions United Rake, Peddle Cheap, Packet Wrench, and Beach Head, all delivered from a FOXACID subsystem called Ferret Cannon." Oh, lord. "Oh, how I love some of these code names," writes Bruce. "On the other hand, EGOTISTICALGIRAFFE has to be the dumbest code name ever.

"Snowden explained this to Guardian reporter Glenn Greenwald in Hong Kong. If the target is a high-value one, FOXACID might run a rare zero-day exploit that it developed or purchased. If the target is technically sophisticated, FOXACID might decide that there's too much chance for discovery, and keeping the zero-day exploit a secret is more important. If the target is a low-value one, FOXACID might run an exploit that's less

valuable. If the target is a low-value and technically sophisticated target, FOXACID might even run an already-known vulnerability. We know that the NSA receives advance warning from Microsoft of vulnerabilities that will soon be patched. There's not much of a loss if an exploit based on that vulnerability is discovered." Meaning that the window of opportunity will be pretty short. "FOXACID has tiers of exploits it can run" - tiers - "and uses a complicated trade-off system to determine which one to run against a particular target.

"This cost-benefit analysis doesn't end at successful exploitation. According to Snowden, the TAO - that's Tailored Access Operations - operators running the FOXACID system have a detailed flowchart with tons of rules about when to stop. If something doesn't work, stop. If they detect a PSP, a personal security product, stop. If anything goes weird, stop. This is how the NSA avoids detection, and also how it takes mid-level computer operators and turn them into what they call 'cyberwarriors.'"

Leo: It's interesting because that same exact behavior sounds very familiar. Remember, was it Flame? What was the name of the virus nobody ever claimed but we always suspected was written by a governmental agency? Did exactly the same thing.

Steve: It was very cautious, yes.

Leo: If it detected this, stop. If it detected that - so, boy, that's interesting. You have to wonder - well, anyway.

Steve: And he says: "It's not that they're skilled hackers, it's that the procedures..."

Leo: Stuxnet? Was it Stuxnet? All right.

Steve: "It's that the procedures do the work for them." So they literally follow a flowchart and just step by step doing what they need to.

Leo: Just script kiddies.

Steve: "And they're super cautious about what they do. While the NSA excels at performing this cost-benefit analysis at the tactical level, it's far less competent at doing the same thing at the policy level. The organization seems to be good enough at assessing the risk of discovery - for example, if the target of an intelligence-gathering effort discovers that effort - but to have completely ignored" - and this is Bruce's whole point of this article - "have completely ignored the risks of those efforts becoming front page news. It's not just in the U.S., where newspapers are heavy with reports of the NSA spying on every Verizon customer, spying on domestic email users, and secretly working to cripple commercial cryptography systems, but also around the world, most notably in Brazil, Belgium, and the EU, the European Union. All of these operations have caused significant blowback for the NSA, for the U.S., and for the Internet as a whole.

"The NSA spent decades operating in almost complete secrecy, but those days are over.

As the corporate world learned years ago, secrets are hard to keep in the Information Age, and openness is a safer strategy. The tendency to classify everything means that the NSA won't be able to sort out what really needs to remain secret from everything else. The younger generation is more used to radical transparency than secrecy and is less invested in the national security state. And whistleblowing is the civil disobedience of our time." Finally, he says: "At this point, the NSA has to assume that all of its operations will become public, probably sooner than it would like. It has to start taking that into account when weighing the costs and benefits of those operations. And it now has to be just as cautious about new eavesdropping operations as it is about FOXACID exploits attacks against users."

Leo: Wow.

Steve: Yeah, yeah. So nice, nice summary.

Leo: You know, mentioning the Guardian and Glenn Greenwald, you saw that he's leaving the Guardian.

Steve: Yes.

Leo: But I thought very interesting where he's going. I don't know if you've seen that detail.

Steve: I didn't see where. But he did say, because I think I saw the news before he announced it, and I haven't followed up, he did say any other journalist given this opportunity would take it. So...

Leo: So it turns out it's Pierre Omidyar, the guy who founded eBay and is pretty much one of the wealthiest men in the world. He, according to Jay Rosen, journalist professor, writes about this in his blog, he tried to buy the Washington Post, was outbid by Jeff Bezos, and then started thinking, well, what if I took that same amount of money or more, a quarter of a billion dollars, and put it towards building a new investigative journalistic enterprise?

Steve: Wow.

Leo: And so apparently that's who Glenn Greenwald is going to be teaming up with. He's looking for other well-known names. It's going to be an interesting thing.

Steve: Interesting. We'll be talking here, we have a piece about a recently gone open source system called SecureDrop we've talked about before. But first I want to mention something that I think is really just good news. We've talked about Matthew Green, who's the cryptographer at Johns Hopkins. We've been talking about him a lot because he's been involved in blogging recently. He's one of the people behind the TrueCrypt audit, which is a tremendous effort. He's created IsTrueCryptAuditedYet.com.

And the top of the IsTrueCryptAuditedYet.com page says: "TrueCrypt is an open source file and disk encryption software package used by people all over the world. But a complete cryptanalysis has not" - never - "been performed on the software, and questions remain about differences between Windows, Linux and Mac OS X versions. In addition, there has been no legal review of the current TrueCrypt v3.0 open source license, preventing inclusion in most of the free operating systems including Ubuntu, Debian, Red Hat, CentOS and Fedora. We want to be able to trust it, but a fully audited, independently verified repository and software distribution would make us feel better about trusting our security to this software. We're pledging this money to sponsor a comprehensive public audit of TrueCrypt." So right now they have - their goal is to raise $25,000. They have 16 of that 25, although a $10,000 lump came from an Atlanta-based security firm.

Matthew Green wrote in his own blog: "We're now in a place where we have nearly, but not quite enough to get a serious audit done. That depends on how many favors we can get from the security evaluation companies. I'm trying to answer that this week." And then in his blog he also wrote: "In case you haven't noticed, there's a shortage of high-quality and usable encryption software out there. TrueCrypt is an enormous deviation from this trend. It's nice; it's pretty; it's remarkably usable. My non-technical lawyer friends have been known to use it from time to time, and that's the best 'usable security' compliment you can give a piece of software."

He said: "But the better answer is because TrueCrypt is important. Lots of people use it to store very sensitive information. That includes corporate secrets and private personal information. Bruce Schneier is even using it to store information on his personal air-gapped super-laptop, after he reviews leaked NSA documents. We should be sweating bullets about the security of a piece of software like this."

So, I mean, and this has come up often. There's, like, there's weird conspiracy theories about, like, the past domain registrations of the TrueCrypt domain. And I see stuff like this from time to time, and it's like, well, okay. But this is just good news. Everyone on the podcast knows that I'm a huge fan of TrueCrypt. I really think that Matt is right, that this - it beautifully combines ease of use and really good security. But it needs an audit. And I mean, and this is always what you and I talk about, Leo. It's one thing for it to be open source. But if no one independent goes through and reads the source, I mean, who really understands cryptography, then it is possible that it could have some little widgets snuck in.

In fact, we've got a story about that coming up about a D-Link router hack. But so TrueCrypt needs this. It needs an audit, and then that audited code needs to be locked down, and any change made to it similarly looked at very carefully so that we can then say, okay, we have absolute, I mean, we trust it because we believe in the intentions of everyone who is involved. But we don't - we can't really assert to absolute certainty the intentions of everyone involved. We just assume good, well-meaning people built this for us. So...

Leo: Yeah, I mean, I've always kind of presumed, because something is open source, that somebody, at least informally, is looking at it. But I guess it's the case that there's a lot of code there and it's possible there's something snuck in. It'd be hard to do. Pretty hard to do.

Steve: Yeah, I don't think anyone has probably looked at it, Leo.

Leo: Well, people look at it all the time. No, no.

Steve: Well, but, yeah, but, I mean - okay. Who?

Leo: Interested parties.

Steve: The good news is we're going to absolutely have a certificate.

Leo: I mean, you can download all the code yourself and put it on your computer and comb through it. I would be - I would be surprised if many people have not done so. Maybe we're all assuming the other guy's doing it. I don't know.

Steve: I think that's what happens. Yeah.

Leo: Yeah. It certainly wouldn't hurt to have an official audit, that's for sure.

Steve: I think it's great. I absolutely think it's, I mean, the problem is we want to relieve people of their concern. We want to be able to say it has been audited by security-aware people, end of story.

Leo: Right.

Steve: And, I mean, it makes sense to do that for a super-popular, heavily used piece of software. I mean, TrueCrypt's what I use. No computer of mine has ever been hacked or attacked, but it just makes sense to have that. I wouldn't operate a laptop without it because laptops are mobile by nature, and stuff happens.

There is a really interesting project, SecureDrop, which has just been moved over to GitHub. If you go to GitHub, you can - it's a - Freedom of Press is the account, and SecureDrop is the subdirectory under that. It is an open source whistleblower submission system which has been managed by Freedom of the Press Foundation. And the idea is that media organizations can use it to securely accept documents from anonymous sources. The code was originally written by Aaron Schwartz. We've talked about this a couple times. It is a complex system; but in order to do this, unfortunately, it's going to have to be complex. So there are five machines, sort of five machine roles in the system. And all of them - well, okay, four dedicated machines, all of which would be located in the offices of an organization that wanted to securely accept documents.

So, for example, in this new organization where Glenn Greenwald is going to be, this is what they would use. They would set up these - and it's all open source, using versions of Linux. They've got a USB-bootable system called Tails OS. And so there is a - one of these computers is a so-called "viewing station" machine, which should be an air-gapped laptop, meaning no wire networking connected to it, an air-gapped laptop running the Tails OS from a USB stick. And in the description it says "that journalists use to decrypt and view submitted documents." And it says, "If this laptop does not have a DVD drive,

buy an external DVD drive you can use with it." So that's one machine is an air-gapped laptop.

Then there will be what they call the "source server," is an Ubuntu server running a Tor hidden service that sources use to send messages and documents to journalists. Then there's a "document server" machine, another Ubuntu server running a Tor hidden service that journalists use to download encrypted documents and respond to sources. And then there's a third Ubuntu server which is a so-called "monitor," which monitors the source and document servers and sends email alerts. So it heavily uses Tor hidden services, and the idea being that independent sources can access these machines through Tor, which gives them anonymity, and there's a broadcasting mechanism where credentials are made available and usernames are kept anonymous that allows anonymous people to securely encrypt and upload through Tor that end up arriving at these systems, where then the encrypted document is lifted from the document server and hand-carried over to, probably burned to DVD.

I haven't gone into the protocol, but that's probably why this viewing station, it boots from USB, but you probably burn the document to DVD, take it over to the air-gapped laptop, which is able then to decrypt with the recipient's credentials in order to make the documents available. So, yes, it's a lot of moving parts. But it's been very carefully designed in order to achieve the goal that everybody's identity is protected, the information is completely safe, yet you are able to have a conversation, ultimately, between a source and a set of journalists.

Leo: Interesting, yeah.

Steve: So a very cool technology, yeah. And last bit of news for the week. Like I said, we had a lot. I tweeted this so people could get it from the tweet. I also created an all-lowercase bit.ly link. So it's bit.ly/ and then just this episode. We're Episode 246, so it's sn-246. So it's a bit.ly link, bit.ly/sn-246. It is a wonderful walkthrough of a guy who hacked the firmware belonging to a family of D-Link, standard D-Link consumer routers and found a backdoor. He wasn't looking for it. He just fired up some music that he describes at the top of his posting, Rush I think it is, I've never heard of them, but that's, you know, Sarah probably has.

Leo: You're never heard of Rush? Come on. Okay.

Steve: No.

Leo: A fine Canadian rock band headed by Geddy Lee.

Steve: Good. So he had good music to help him with his hacking. And what he goes through, he just - he, like, looks through a list of all the subroutines, and he sees the authenticate subroutine. And so he says, oh, that's interesting. Let's go see what the, well, I'm just curious, what does the authenticate algorithm look like? And he's looking at ARM assembly language using a disassembly and reverse-engineering tool to see what the branch conditions are and string comparisons. And he sees an interesting string comparison that for some reason in the authenticate routine he follows some pointers, and it seems to be comparing the contents of the user agent. And he thinks that's odd.

You'd want to be checking the password and the username. Why would you care about the user agent, which of course is, like, the user agent is Firefox or Internet Explorer or Safari or whatever.

So he follows it along some more, and he finds this weird string which the user agent, the contents of the user agent header in a query is being compared with. And it's compared with a string which has a weird-looking text - it's roodkcableoj28840ybtide - until you reverse that string. And that string in reverse reads editby04882joelbackdoor.

Leo: Wow. Is that a date, you think?

Steve: I don't know what the 04882 is. Maybe it's something in leetspeak if you turn it upside down or backwards or who knows. But it turns out this is a backdoor for logging into a family, a broad family that all use this firmware, of D-Link routers. And all you need to do is change your user agent to that string, and you do not need a username and password. It bypasses authentication on these D-Link routers.

Leo: Wow.

Steve: And it's like, whoopsie. Now, hopefully this is not exposed to the WAN. Everybody should have certainly turned off WAN-side login on any router. I mean, that's Security 101. Unfortunately, apparently there's that search engine, I can't remember the name of it now, which does aggregate - it's an Internet search engine for publicly available servers. And apparently there's all - I saw that there was an NMAP script had been written for finding these hackable routers. So it may very well be that we have, I don't know, I haven't looked to see whether there is WAN-side administration available. I just - I hope not because, if there is, and if it's on by default, then this is a massive breach of security for this family of D-Link routers, all using this. But mostly I just, for people who are curious, it's very well written. He shows screenshots of each phase of his discovery, little subroutines linked together courtesy of this great reverse-engineering software. He comments what he finds. Anyway, so it's bit.ly, bit.ly/sn-264.

Leo: 246.

Steve: I'm sorry. Sorry, yes, 246.

Leo: And it's confused people because the episode number of 426. But it is 246.

Steve: Oh, my...

Leo: It's okay. Somebody in the chatroom also added 426, so both work.

Steve: Oh. Sorry about that, folks, yes.

Leo: Doesn't matter. You said that right. It's driving people in the chatroom crazy. They're apparently fairly anal. No, no, Steve, the episode number is 426.

Steve: Okay, yes. So...

Leo: Except it doesn't work. Moving along. Both work now.

Steve: So total - we're out of news. A little bit of miscellaneousness. I saw a tweet from someone who's, well, his name is Pudding, and his Twitter handle is @tonofpudding. So I don't know what that means.

Leo: Well, it's more pudding than just one pudding. It's a lot of pudding.

Steve: It's a lot of pudding, yeah. So he said, @SGgrc, he said, "I bet Elaine is now a total security expert, having to listen closely to every episode to transcribe them." And what's funny, the thing that caught my eye is that Elaine just said last week - see, she does amazing due diligence for the transcripts. So when we were talking about Taylor at Defuse.ca last week, and you knew of Defuse.ca and remembered who Taylor was, well, Elaine googles, goes to his site and looks around.

Leo: Wow. She verifies. Wow.

Steve: Because she wants to spell his name and make sure she gets the URL right, I mean, she really does it.

Leo: You go, Elaine, wow.

Steve: So apparently in Taylor's rsum he says, one of the items, I guess it's somewhere on his site because Elaine found it, he says "Listened to every SN podcast twice." And Elaine read that and said - so wrote all this to me and said, you know, so have I.

Leo: Come to think of it.

Steve: Because she listens to it once to transcribe, and then she proofs it. She listens to the entire thing a second time, proofreading her first pass transcription. So she said, yes, Taylor and I have listened to every Security Now! podcast twice.

Leo: And she is now officially a security expert.

Steve: So, okay, Leo. I had not seen "Gravity" last week.

Leo: Oh, you didn't go - oh, you know what, I didn't get to see it. We said I would, but I haven't yet.

Steve: Well, I have.

Leo: Okay. No spoilers.

Steve: No, no spoilers, of course. I would just say...

Leo: It has something to do with space and falling, I think.

Steve: Okay. Well, then, we'll leave it at that. I liked it. I wouldn't really call it science fiction. I would call it science drama.

Leo: Yes.

Steve: And it was fun. It was fun science drama. I mean, it was - but if you're going, if you want sci-fi, "Ender's Game." That's what we're waiting for.

Leo: Yes. Exciting, yeah. But read the book first.

Steve: Yeah. You know that I did reread the book, just because I knew that now the movie was coming, and I wanted to - that's Orson Scott Card, for those who don't know.

Leo: Read the book first.

Steve: And we do have "The Tomorrow People." I watched it. I watched the first episode last Wednesday.

Leo: What did you think?

Steve: Eh, I mean, it's a little bubblegum. But beggars can't be choosers, and I'm definitely begging for sci-fi. So I'll give it a while and hope. It looks like it could be fun. We'll see how it evolves.

Leo: Nobody has yet disproved my theorem that science fiction on TV is pretty much not so great.

Steve: Well, "Firefly" was great.

Leo: "Firefly" was, no, you're right. Okay, I'll give you that.

Steve: And "Star Trek." "Star Trek" all began on television. I would say there is no good contemporary - there's nothing now at the moment that is...

Leo: "Firefly" was the last. And it was mangled on TV. So it doesn't complete disprove my theorem.

Steve: Good point. They did it out of order, and they canceled it without even airing all the episodes they had made. Like, what is wrong with you people?

Leo: It was made properly, just never displayed on TV properly. Just briefly, a couple of weeks ago while I was on vacation, Steve introduced something called SQRL, Secure QR code Login. And the world is beating a path to your door, it sounds like.

Steve: Well, I've been watching a lot of the dialogue going on. And the short version is, yes, this concept has captivated a lot of people. What I like about it is that it is simple. And so the GRC newsgroup has gone crazy. I'm something like 800 posts behind, and I was current for a while. But I think we're about 2,200 postings that have been made, just because so many people want to be involved and to communicate. As I mentioned last week, I've been contacted by the W3C, the World Wide Web Consortium, about considering adding this to the HTML5 spec, opening a dialogue. Next week is the HTML5 Developers Conference, the web developers conference at Moscone Center. And a Dan Holmlund is going to be giving a presentation on SQRL, basically doing a presentation to explain...

Leo: Oh, awesome.

Steve: ...what it is and how it works.

Leo: Wow.

Steve: In his little snippet, in his synopsis of his presentation, he said: "Recently, the computer security community has been set on fire by a proposal for a new authentication scheme named SQRL from Steve Gibson of GRC. SQRL is an easy-to-use, high-security replacement for usernames, passwords, reminders, one-time-code authenticators. It provides authentication that can be anonymous, and it requires no trusted third party that can be compromised by hackers or three-letter government organizations. We will walk through how SQRL works and how it can be implemented on your web or mobile application."

So, I mean, and if you look, there's an Implementations page in the SQRL pages at GRC, as you mentioned last week, GRC.com/sqrl. Actually, when you mentioned it, that link wasn't working because it's actually sqrl/sqrl. But I thought, ooh, I'd better fix that, so I

did.

Leo: Oh, good.

Steve: So that works now.

Leo: Too many SQRLs. That can always screw things up.

Steve: But, I mean, people are, like, writing code like crazy. So I've sort of been trying to ride herd of this, like, too much interest, almost. And the thing I've run across is - and this is not a complaint, I consider this good, it's just a lot - is everyone wants it to do everything. And it does not do everything. And what I believe we have with it is there are many benefits. One of them is its simplicity, the fact that it isn't a kitchen sink, that it doesn't involve a third party. There's plenty of identification schemes trying to even exist that are third-party schemes. But I think third-party schemes, all of them, are dead now in this post-NSA, post-Snowden era.

So my best example of a system that works, but it's not perfect, and I've used this, and it's one of my favorite analogies, is cars, driving cars. Imagine that cars didn't exist, that there hadn't been an evolution, but that they just, like, people were saying, okay, we need to stop walking everywhere. What are we going to do? And someone says, I know. Let's create a 2,000-pound, gas-powered, high-speed missile with a steering wheel, and we're going to just let people go wherever they want to. Well, I mean, people would have said, you're crazy. That's insane. People are going to crash into each other. Oh.

Leo: Oh, the horror.

Steve: Well, I guess that could happen, but most people are not suicidal. And we'll help them. We'll give them traffic lights. And everyone will agree, if the light is red, it would be good to stop your missile before crossing oncoming traffic. Everybody will be happier if we agree to some rules.

Leo: Yes.

Steve: And so my point is I don't think it could have ever happened.

Leo: No.

Steve: If, I mean, it just would have never happened if you didn't already have it. It's because we slowly evolved from horses - in fact, one of my very favorite Henry Ford lines is - it echoes a Steve Jobs line. Steve Jobs said, I love this, "It's not the customer's job to know what they want." That's just brilliant. I love that. He didn't do focus groups. He just said, "Here it is. Suck it up." And of course we all sucked up all of Apple's goodies.

Leo: Right.

Steve: Henry Ford said, "I asked people what they wanted, and they said they wanted a faster horse." So...

Leo: I hate to say it because I love that quote, and I've quoted it. And I have been told that that's - that he never really said it. But it's...

Steve: Henry Ford?

Leo: Yeah, and it's a great line. I love the line, yeah.

Steve: Great quote. Anyway, so my point is that what I've been trying to do is keep the system simple. For example, revocation. The idea is people - I say, okay, you just have one master key, and we're going to help you protect it. I mean, we're really going to go out of our way to help you, to support the idea of one master key. The beauty is that the key is never used. Nothing, there's no vulnerability to using it. Websites don't get it. They get a derivative of it based on their domain name, which allows you to identify yourself to them. And even if they leaked that derivative, your identity for them, it's different on every other site. So people just - they love all of that. They say, oh, okay, great. But then they say, but what if I lose it? Okay, well, right. What if you run through the red light?

So my point is that we will do everything we can to protect you from yourself, to help this simple system actually work. So there's two aspects to it. To keep bad guys from getting it, we use extremely strong encryption. And there's not really even a term. I say "deeply encrypted" when I talk about, well, it takes a minute to do the encryption, which allows you to then export your key in a QR code, like to print it out, to stick it in a drawer so that you're prevented from losing it. And it also takes a second or two when you enter your password to remind the phone or to prove that you're the one using it.

So we really create barriers to you losing it, that is, getting out of your control, or bad guys getting it, because we want to make it as practical as possible just to have one key. If we can, then the benefits are huge because you can literally use this to identify yourself uniquely across the entire Internet in a way where you don't have to have any per-website stuff. It's just this one thing.

But people said, okay. What if I have a bad breakup with my - with a partner. And he or she has my key in their phone. And you can have your key in someone else's phone, protected by your password, and it's safe because we've made it so difficult to do password guessing. Or what if your phone is confiscated at the border and they keep it for a week before giving it back to you? So the point is there are arguably some scenarios where you could forever after something happens feel uncomfortable that your key may have gotten out of your control. Somebody has it on their phone; or they decrypted your phone, the government decrypted your phone and then sucked its data out and gave it back to you, and now they can be working on it in the background, where again, all the protections we have prevent it from immediate - prevent your key from having any rapid reverse-engineering, no way for anyone to get it without, I mean, it's completely infeasible to do brute-force attacks. But still, what if?

And so we've come up with a small improvement, the simple improvement to the protocol which solves this problem. And that is that the inevitable problem, or the possibility - we don't want it to be like something everyone does. But it's like, okay, what if you wanted to retire a key for any reason? All we've changed is you can take whatever your current key is and assign it as your previous key, sort of just change it to your previous key, and invent a new one. So now your SQRL client has the notion of a previous key. Just one. Again, you don't need 10 because this is the kind of thing that shouldn't ever happen. But if it does, then we want to provide a means for replacing a key that you regret may have gotten out of your control somehow. And even though it doesn't mean you're hackable, it's just we ought to have a way to do that.

Now, I've always felt that websites, a website that you're using, would give you means to manage your account. They probably still have an email address for you. I mean, that's still something you uniquely control. And so you probably could go to a SQRL-supporting site and say, "Hi there, it's me, I want to change my email address," the way we do now. And they might say, okay, well, we don't have a password on file for you because you're using SQRL. But we did ask you for some security questions so you can prove who you are through another means. I mean, these sorts of things still function.

Well, certainly there ought to be a means there to say "I would like to replace my key." I mean, I have always imagined you'd be able to do that. The problem is it's still kind of awkward. It's like, well, you've got to log in with your old key. And then what, do you have, like, how do you - you have to switch users, your user account in SQRL in order to say, okay, here's - now I'm going to scan my new key, so you get that. Anyway, that's just sort of awkward.

So we've made it simple to do this. Anytime your SQRL authenticating device, your smartphone or a SQRL app on a laptop or a computer, however you're doing it, anytime you have defined a previous key, your old key, then the authentication query, which is what identifies you, simply provides both. It provides your new key and your old key and two signatures of that. So the two signatures is the old sig and the current sig, essentially.

So what that does is, anytime you log into a website, it will - and the website sees you have an old key defined, you're giving it an old key, it will first check to see whether it has a record of you under the old key. If so, it replaces that immediately with the new key and forgets the old key. And it knows that you were the owner once of the old key because you have correctly signed the old key. Actually, the old key's private key was used to sign this whole query, as was the new key's private key. So basically you've double-signed this to say I know both of the private keys that are associated with this old identity and the new identity. And the site simply replaces the old one with the new one.

So logistically that makes this arguably important sometimes or possibly need to relatively easily replace your keys manageable. You simply retire the key you were using, come up with a new one, and then you just go, you just visit the various important sites that you use most, your bank and Amazon and Google and those major ones. And your prior identity is flushed from them. There will certainly be some, like random blogs and things, that you don't get to immediately. They'll still have your old identity until you next go there, and then your identity will automatically be replaced on the fly for you.

So that's a simple change, an enhancement we made to the protocol that solves the problem of, if I lose control of my galactic master identity key, and I really, really, really have to change it, how do I do that? And it's not like we changed every key in the world. There have been proposals in GRC's newsgroup, like keeping a record of every site you log into. Well, okay. Then we have a very stateful client. Then how do you move that

between devices? What if you log in somewhere else on a device that isn't synced to the cloud? I mean, suddenly the whole system becomes a huge problem, if you try to perfectly solve the revocation problem. I argue, just as with driving cars, if we provide people with the tools that they need so that they will not hurt themselves, then this operates in a sweet spot where it is very simple, easy to implement, and provides tremendous value. And we help people to be responsible.

And then the second thing which has happened in the last week which is very cool is we actually have come up with extremely good anti-phishing protection. Phishing is a topic we've discussed on the podcast through the years. It's an ongoing problem for the Internet. Typically you get a link in email, and it's because of this problem that the wisdom is never click a link in email. It's because you can't see where the link is taking you.

And it may look, it looks like a PayPal invoice. I've seen some spam like that. It's like, hey, we just wanted to let you know we just cleared the $374 purchase you made, yay, in PayPal. Click here if you want to check your account. And so that's of course a scare tactic. Someone says, wait a minute, I didn't authorize a 300 and whatever it was dollar purchase. So they click the link, and it takes them to a page that, sure enough, looks just like PayPal, but it's not. It's a variation on PayPal's name, something where they're just not going to notice that it's .cn instead of .com, for example.

So people have said, hey, Steve, is there anything that SQRL can do to solve this phishing problem? It turns out there is because, first of all, a site that wants to spoof SQRL login has to go to a much greater effort to do so. In the PayPal example, it's a passive spoof. They show you a PayPal page that looks just like PayPal and says give us your username and password. And you type that into this fake PayPal page and hit Submit. Instantly they have your credentials. They've got your username and password for PayPal. And now you're in trouble. In order to successfully spoof login with SQRL, the spoofed site would have to get a valid SQRL code from PayPal.

Now, that can be done. Basically it becomes a man-in-the-middle attack where - so that ups the ante, just using SQRL ups the ante of phishing, but still makes it possible. So that site, when you display the page, behind the scenes the evil site goes and gets a - essentially it starts a logon on PayPal, gets the SQRL code that PayPal thinks it's showing you, and then the bad site displays it on its page. Now, you scan that. And what you're doing is, because it's a true PayPal link that just came from the PayPal logon page, you are truly, you are authenticating yourself to PayPal, but you're authenticating the session which the evil site started. That is, it opened the login page and then showed you the SQRL code, which you would be authenticating.

The problem is the IP addresses don't match. Think about it. The web server, .cn or .ru or wherever it is, the web server asked PayPal, the PayPal server, from its IP for the SQRL code. When you authenticate, you're coming from either your web browser's IP or from your phone. They will be different. And so there is a large class of phishing which we're able to block. Remember that we talked last week about being able to either optically scan the QR code to use the whole multifactor "I'm using my phone as my identity" approach. But many people, even Tom, when I first mentioned this two weeks ago with Tom, he said, well, what about if I just - if I've got my laptop with me, and I want to log on with my laptop? And of course since then we have fully fleshed out that.

You can have an SQRL client installed on your computer, whether a laptop, a desktop, a tablet, or your phone. And if you're logging in on that device, then you just tap or click on the SQRL code. And because it has an sqrl://, it understands that that's the so-called scheme of the URL and does the authentication for you. In that case, in any instance

where you're logging in on the same device that you're browsing, the IP address will be the same. Your browser will have asked for the login page, and that will be the same IP as the SQRL client asks for, performs its authentication query.

So what we've added to the spec is something very simple. The client knows if the IP address is expected to be the same. It's probably not the same if you're using an optical scan and a cellular carrier for bandwidth because then the IP is going to be different than the machine you're scanning. But if you were at home, and your cell phone was on your home WiFi network, then your public IP of your browser and your phone are still going to be the same. So even there, even using your phone with the same WiFi on the same network as your computer, the IP that the public sees is the same.

What's cool is if the client knows if the IP is expected to be the same. If it's a client on a laptop or desktop or tablet or phone, not being optically scanned, and not cellular carrier, it adds an option to the query. The option is "Enforce." And what this does is it tells the web server where you're authenticating to enforce a same-IP policy. And the option cannot be removed by any man in the middle because the signature which we use to sign the URL encompasses all of the options and other features. So we're signing the whole package. So the signature, after the signature gets verified, the web server sees that we've asked for enforcement; we, the client who's doing the authentication, is asking for enforcement of the same-IP policy. And if the IP that it gave the SQRL code to is different from the IP that is asking to authenticate that SQRL code, it fails. It sends an error message back saying "IP mismatch, authentication failure."

And so what that does is it essentially imbues this system, not in every possible case, not in the optical scan cellular carrier because there you'd expect your IP of your computer and your phone to be different. But in the huge instance of anyone working at home, at office, where they're using their same SQRL identity to log into websites during the day, or if you're on a tablet, or if you're logging in on your phone, in all of those instances the IP should be the same. This completely detects and blocks phishing, which nothing else has ever been able to do before.

Leo: Seems that, though, in a lot of cases, that this is an option that's going to be left off. Is it going to be off by default? Probably; right?

Steve: On by default.

Leo: On by default.

Steve: Yeah, it'll be off for the optical scan mode, but it'll be on for the authenticating non-scanning when you're clicking or tapping because the IP ought to always be the same. You'll be able to override it. So essentially it'll come, your authenticating device, whether your phone or the SQRL client, will come back and say, explain in understandable English for our moms that, well, I don't know how we're going to explain it to our moms. But you'll be able to push past that if you want to. It'll explain that there may be a problem with the website, that there is an authentication problem, refresh the page, make sure the URL is spelled correctly, whatever we end up, how we choose to end up wording this. But it is always the case that the IP should be the same. And only in the event of a third-party obtaining the SQRL code on your behalf and trying to trick you would the IP mismatch. So, and again, we'll put this out; we'll experiment with how it works and see how it goes.

Leo: Yeah. I mean, if it becomes an issue, you can obviously - nothing's written in stone.

Steve: Yeah, you'll be able to turn off the checkbox.

Leo: Right, right. Very interesting. Once again, more strides going forward. So tomorrow is the HTML5 presentation? Is that tomorrow?

Steve: No, it's next week.

Leo: Next week.

Steve: It's next week.

Leo: Very interesting.

Steve: Yeah. At HTML5devconf.com.

Leo: That's awesome.

Steve: Is the location.

Leo: Yeah.

Steve: So, yeah. I'm not going to turn this obviously into the SQRL podcast. I hope this - I'm going to try to - we'll do a Q&A next week and take a lot of questions, and I will certainly not have SQRL dominate the podcast. But that's what I'm working on until we get it done.

Leo: Cool. If you want to participate, of course, there's a group, working group going on right now at Steve's site, GRC.com, in the forum there. You can also submit questions for next week on any topic, SQRL or otherwise, at GRC.com/feedback. GRC is where SpinRite lives. You didn't mention SpinRite this week, I don't know why, Steve, but I'm going to mention it...

Steve: Thank you.

Leo: ...the world's best hard drive recovery and maintenance utility. If you have hard drives, you must have SpinRite. You can get it right now at GRC.com, along

with a bunch of other...

Steve: It does pay all the bills.

Leo: ...freebies. Yeah, that pays the bills.

Steve: And a free upgrade to the next version is guaranteed.

Leo: Yeah, yeah. That's awesome, too. He's working on that, too. GRC.com is where you'll find 16Kb audio of this show for the bandwidth-impaired. Those full transcriptions we mentioned by Elaine Farris, very nicely done. We have full bandwidth audio and video of the show at our site, TWiT.tv/sn, for Security Now!. Collect all 462 or whatever, 26. I mean, you might as well get them all. Must have them all. They're all there. And have a complete set, just like I see you have the Oxford English Dictionary. You know, I never noticed that over your right shoulder, yeah, your right shoulder.

Steve: Oh, yeah. Love it, wow.

Leo: And somebody's mentioned that your blinking lights are not moving as fast. Did you slow down the refresh rate on your PDP-8?

Steve: Ah, we've got smart, we've got...

Leo: They pay attention.

Steve: We've got sharp-eyed people watching.

Leo: They pay attention.

Steve: I changed - the switches allow me to change the characterization. I thought, oh, let's change the way it feels.

Leo: Little less frenetic.

Steve: Yeah.

Leo: If you want to watch the show live, we do it Wednesdays, 11:00 a.m. Pacific, that's 18:00 UTC on TWiT.tv. Please tune in live and watch. The chatroom, of course,

is always a big part of all of our shows, and that's the only way you can be a part of the chatroom is to watch. But you can also visit us in-studio. We have some nice people visiting this week. All you have to do is email tickets@twit.tv. There are really limited spaces, only about five seats here in the little studio. Big studio has kind of unlimited room. So do email us and let us know ahead of time so we can make sure we can get you in: tickets@twit.tv. Steve will be here next week to talk more about security and answer some questions, I think.

Steve: Yup, GRC.com/feedback will get you to the Security Now! feedback page, GRC.com/feedback.

Leo: Thanks, Steve. We'll see you next time on Security Now!.

Steve: Thanks, Leo.