Listener Feedback #221

Description: Leo and I discuss the week's major security events and discuss questions and comments from listeners of previous episodes. We 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-531.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-531-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. He's going to answer 10 questions. He's got the latest security news. I don't even have to sell this show to you. I know you listen every week. Good news! You've got another fresh Security Now! coming up next.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 531, recorded Tuesday, October 27th, 2015: Your questions, Steve's answers, #221.

It's time for Security Now!, the show where we protect you and your loved ones online with the Explainer in Chief, Mr. Steven "Tiberius" Gibson. He's right there, sitting next to my pumpkin likeness. Happy Halloween.

Steve Gibson: Yes, this is our - well, it's not really our Halloween episode. But by the time we get to the next one, it's be behind us.

Leo: It'll be over, and then we can breathe a sigh of relief.

Steve: Oh, yeah. So Q&A today.

Leo: Yes.

Steve: Our 221st Q&A. We're going to talk a little bit more about 1Password. I did some more research, and I now understand what it was that happened with this metadata leakage, that is, what critical design mistake they made, which they basically sort of have

been recovering from ever since. We'll talk about that. But I have to say I came away impressed with what I saw from what they had, though still not really with their conduct, which was my major complaint last time.

I had intended to talk about a really interesting research paper about some security researchers plowed into some Western Digital external encrypted drives and came away unimpressed. But when I got into the depths of this, there was so much there, and it was so juicy, that I thought, oh, this is an episode all by itself. So three episodes from now, that's what I've got slated, because next week we're going to do a real stem-winding propeller-head episode on the whole dynamic call structure of iOS that we talked about, the whole problem of them vetting applications through the App Store. What I said was correct. But at a recent conference, just a couple weeks ago, there was a beautiful paper on exactly this topic. So that'll be next week.

In a lot of news has been this concern about, essentially, how the NSA is seeing into our encrypted data. That had to have, like, maxed out the Twitter stream in terms of people picking this up and forwarding it to me, so we'll cover that. An update on the Let's Encrypt project that has another milestone. Some news on the ever-beleaguered SHA-1 hash and its history. We'll come back to that because we talked about it two weeks ago. We've got miscellaneous stuff to talk about. And a Q&A, so 10 questions, thoughts, and observations from our terrific listeners.

Leo: A massive program.

Steve: So the picture of the week, I had to do a double-take on this, to convince myself it was not a spoof. But there's a source: "Navigating the Digital Age: The Definitive Cybersecurity Guide for Directors and Officers Provides Actionable Advice and Best Practices." And this picture has the caption: "How the New York Stock Exchange says companies should decide whether to disclose hacks."

Leo: [Laughing] It's a decision tree.

Steve: Yes. And it's somewhat chilling, actually. If you go through this, it's like, sort of there in the middle, "Will you disclose anyway," so if you discover a hack - well, okay. So...

Leo: You might as well walk us through this tree.

Steve: Is it material? So if it's yes, then it's an immediate arrow down to leaning toward disclosure.

Leo: And specifically a legal form of disclosure, an 8K disclosure.

Steve: Right. If it's not material, no, then we go to, is there a separate obligation to disclose? And so, if yes, then once again lean toward disclosure.

Leo: Yeah.

Steve: If no separate obligation, then we go to, will you disclose anyway via website to third parties, et cetera? If yes once again, back over to leaning toward disclose. If not, then the question is, is discovery - so if you will not disclose anyway via the website or a third party, et cetera, no. Then is discovery of the breach by government or public likely or inevitable? And if you follow the no, then it says, really? Are you sure?

Leo: Really? Sure? Are you sure? Come on, really?

Steve: And so if discovery is likely, yes, then you should disclose. If you're not sure, like if it's like maybe not, then again, better safe than sorry, disclose. But if you're really sure that discovery is unlikely, then we say, yes, we're really sure. Then we go to is there a potential regulation FD issue? I don't know what that FD means. But, if yes, then once again, you'd better get it off your chest. Otherwise, no potential regulation FD, whatever that is. So then we go to will the disclosure itself harm the company? If yes, then finally we get to the other - this is our first visit to the "no" result, lean against disclosure. If the disclosure will not harm the company, eh, might as well go ahead and tell everybody. So lean toward disclosure. If we're not sure if it'll harm the company, then will it compromise security?

Leo: Nah.

Steve: If so, don't tell anybody. If not, then will it trigger securities or other litigation?

Leo: Oh, lord.

Steve: Or investigations. If not, eh, go ahead and talk. If so, don't tell anybody.

Leo: You know, the good news is this kind of really does tell you the thought process that most companies, but especially financial services companies, will go through.

Steve: Yes. I think it's exactly right. It balances, it's a way of working through the tradeoffs and balancing, you know, the likelihood that others will discover it, whether or not anyone already has, whether it's public or not. If it's not discovered, is it likely to be discovered? And, if so, what are the consequences? You know, what are your legal obligations and so forth. So, yeah, it's just - it's a kick.

Leo: FD is probably Fiduciary Duty.

Steve: Oh, I'll bet that's it. Regulation Fiduciary Duty issues, yup.

Leo: Is that right?

Steve: Anyway, I got a kick out of that.

Leo: Fair disclosure. I have an attorney here, and he is, he's whispering in my ear right now, it's fair disclosure.

Steve: Nice. And actually that makes total sense for one of these little boxes.

Leo: It makes better, actually makes more sense, yeah, yeah. So that's a legal responsibility of some sort.

Steve: Very cool.

Leo: Yeah.

Steve: Okay. So the sentiment I shared last week about 1Password upset 1Password users. Not surprisingly. And so I got a lot of venting on Twitter and in direct messages. And, I mean, and I understand. I was tough on 1Password, mostly because it seemed that their priorities were wrong. And that was ultimately all I was arguing with. But people don't want me to disagree with their choice that they've invested in, so I get it that people were upset.

But some people sent me a link to the technical disclosures that 1Password has published. And they are quite forthcoming with all of their docs. And of course we know that's an absolute requirement of somebody who is going to be hoping to provide us with a security product, if they want anyone to really make a knowledgeable appraisal. And I was impressed by that. I was impressed by their style and by their disclosure. I remain unimpressed by just their handling of this.

For example, they did a blog post with the headline, "When a Leak Isn't a Leak," and referred to Dale's post, the Microsoft software guy who started this, and the top of whose blog post I shared last week. They did not link to it, so they didn't make it easy to go find it, but they just referred to it. And I got onto this from their tweet, which said, in response to somebody else, and there was @twit was mentioned, and @SGgrc was mentioned in 1Password's tweet response. They said, "It's encrypted by Dropbox password." And it's like, oh, god, guys. Okay, stop talking, you're hurting yourself. You're making it worse.

So apparently their argument was that, yes, the user is responsible, or some third party is responsible, for encrypting the 1Password database in order to keep the metadata from being available, which is not something, you know, you don't want the company you're trusting to be delegating the security without making it clear. I mean, their position is this doesn't matter. This is not sensitive. And they play up the fact that the password itself is always encrypted. But, you know, fine. But the whole point has been that metadata does matter. So they're now scurrying to fix this, to address the problem, and after three years to move people over to the secure solution.

But the piece of information that I didn't have before, that came from looking at the technical specs of the default organization, is that they made a crucial original design mistake for a good reason, but it hurt them. And that was every single item in the vault is contained in a separate file. Now, their logic was that then they can use existing file sync technology to update only the file that changes, and that changed file will just be one item. So when you change your password, that one item in one file will change. Something will notice, some file sync which is sort of unspecified - they just say, oh, that, you know, it'll work with file synchronization systems - will notice that the timestamp has changed and update that one.

The problem is, and many other people have discovered this over time, that sort of approach doesn't scale. And that's what bit them is, if you have 10 websites, okay, 10 files, not a problem. A couple hundred, it starts to be a problem. A couple thousand, oh, my goodness. Because what that means is, and this is what they were saying was it's not so much that they have to - they would have to decrypt every file. They would have to open every file.

And I'm often amazed when I think about the logic that a contemporary operating system has to go through to decide if the process that is making a request to open a file has the rights to do so. The more you know about what it has to go through, the more amazed you are that we ever actually do get a file opened because we have a - the process itself has a set of rights which are elaborate. Then oftentimes in, for example, a file system like Windows, you have inherited rights which inherit from the root all the way down through the tree with parent and child relationships. Then on any one of those nodes you can have individual overrides of specific privileges that affect that file and may, from that point on, inherit downwards, but maybe not. Anyway, I mean, it's just - it's unbelievable. This is what the OS has to do when you say "Open the file." I mean, it goes through - and of course we want that because we want sophisticated security support from our operating system. That's one of the main things that it offers us.

But so now imagine if you have 2,000 little tiny itty-bitty files, and you need to search them in order to find something. And so what they've done is, with the newer format, they fixed that. Each of the items has a UUID, a universally unique identifier. And the UUID format has the first digit as hex, so it's going to be a zero through nine and then A through F. So that gives them 16 different possible first digits of a UUID.

And so now they have 16 files, and they place each item in one of 16 files, based on the first digit of that item's unique identifier. So they've essentially, they've controlled this explosion of itty-bitty files that would all have to be opened up and rifled through. So they sort of have a tradeoff. So they immediately divide it into one of 16, and then that can grow. And in general, given good random numbers, all 16 files will sort of grow at about the same speed as you populate it.

So I understand the problem they had. Unfortunately, the decision they made when they had this wrong one-file-per-entry architecture, was to speed up what happened after the file got opened, rather than consolidating all those little itty-bitty files which they have now finally done. And as a consequence, metadata got loose.

Now, apparently this all annoyed one very active tweeter, who you asked me about before we began recording, Leo, asking if I was going to address Paul Moore's arguments because, I mean, he's been, you know...

Leo: Bugging both of us for all week.

Steve: Yes, he has.

Leo: Well, he really - he's found something here.

Steve: Well, what he's done is he's reminded us of something that we all knew, and we talked about in the past, and it's a nonissue. So his argument is that LastPass is also leaking metadata. And it isn't. He's wrong. Plain and simple. What he found is, and again, even in his own blog posting from four days ago, he acknowledges that this has all been known since 2012. It's like, yeah, and we've talked about it before.

In order for LastPass to show us the cute little favicons that make finding a site so much easier because our eye instantly spots the new Google icon or whatever it is, in order for it to do that, the LastPass client needs to fetch them from the websites. Joe's logic was, if the clients themselves did that, then that represents a clear privacy and metadata leak because the client would be fetching from the site, so the site's cookie would go with it, the site would know who you were and where you were, and this would occur even if you weren't actively visiting the site, but just if it needed to acquire the favicon for it. So instead, Joe made a conscious design decision to proxy for LastPass, corporate, to proxy for all of its users the favicons.

So what happens is the LastPass client, over HTTPS, so it is not - oh, and that was the other thing is that the client, if sites were not using HTTPS, just HTTP, not only would the destination server obtain the user's cookie and IP address and whatever other information it was getting, and just the fact of the query, but if it was over HTTP, there was no protection for that query. So anybody in an open WiFi or monitoring the connection somehow would see that that was going on and would know that you have an association with that site.

So to solve that problem, LastPass is a proxy. The clients generate an HTTPS secured query to LastPass, asking LastPass to provide the favicon. So LastPass, if it doesn't have it, goes and gets it, and then returns it to the client. So those sites, rather than getting this information from all LastPass users that they are that, all the sites see is a query from LastPass with no cookie, no other identifiable information, and then LastPass forwards it back to the client. So it's very much like sort of using a VPN. We're sort of using LastPass as our VPN to get something that would otherwise not be secure, secured. And that's what that is. And we've talked about it years ago, and everybody knows about it. So, and it's not leaking any metadata in any sense, not like 1Password, where the database itself contains, in plaintext, URLs and titles that the users have assigned.

So anyway, that's what that all is. I came away overall feeling, you know, again, understanding better how 1Password painted themselves into this corner, but not really very impressed with the way they handled this. I mean, I get it that their back's up against the wall. I'm sorry about that. But this is the consequence of having made the wrong decision. And for people who want to continue using it, I think that's fine. I know that this has moved some people to the new format, and that's a good thing. They're in the process of supporting that new format on additional clients.

And one tweet interchange that I had with someone said that, for his purposes, all he was waiting for was Android support of the new format. And so once 1Password provides

that, for him and I imagine a lot of other people, they'll be good to go, and then all the metadata will be encrypted.

So I did plan, I had in my show notes here the discussion of the Western Digital hard disk crypto problem, which we will cover in three weeks because it's just, oh, it just, I mean, we've actually created, as a consequence of last week, a new meme for the show. And that is "doing it wrong." People loved the...

Leo: We might have to do a regular update on this one.

Steve: Yeah, exactly. I mean, it just, I mean, it's like TNO. It just sort of hits exactly the right chord of, sorry, you're doing it wrong.

Leo: Doing it wrong.

Steve: So, and people said, oh, could we have more "doing it wrong" shows? And that's like, well, I'll keep an eye out for them, but...

Leo: I don't think you're going to run out of material. I'll be honest.

Steve: No, although the problem is, I mean, to have a "doing it wrong" show, you really want multiple instances of something being done wrong. And last week it just sort of - it just converged. It just sort of, you know, the show assembled itself so that I realized, wait a minute, four of these things are all about people who did it wrong, so let's make that - but anyway, so I'll keep my eye out for that. I don't know yet whether we could call the WD hard drive crypto implementation that, but we'll know in three weeks.

Just the little abstract from this analysis that was done says: "Self-encrypting devices" - and they made up an acronym, SEDs, self-encrypting devices - "doing full-disk encryption are getting more and more widespread. Hardware-implemented AES encryption provides fast and transparent encryption of all user data on the storage medium, at all times." And of course we can think, like, for example, the iOS, and now we have the latest Marshmallow, not Mushroom, Marshmallow version of Android.

Leo: Did you think it was going to be Mushroom?

Steve: No, remember I called it Mushroom the other day.

Leo: Did you? I missed that.

Steve: I knew it was an "M" word, and I said, what, is it Mushroom?

Leo: I would have stopped you, had I heard it. Oh, man.

Steve: Anyway, so now we're getting on-the-fly encryption on our phones. So these guys continue: "In this paper we will look into some models in a self-encryption external hard drive series, the Western Digital My Passport series. We will describe the security model of these devices and show several security weaknesses like RAM leakage, weak key attacks, and even backdoors on some of these devices, resulting in decrypted user data without the knowledge of any user credentials." Ouch.

Leo: Hmm.

Steve: And so we will do that in three weeks, unless the industry and the world tosses us a curveball.

So there was an interesting story that many people picked up on. I'm sure Ars Technica. I don't remember now, and I didn't put it in my notes here because it was sort of a nonevent. Well, actually the EFF talked about it, too. So but again - okay. So the tag that the press picked up on was, is this the way the NSA is decrypting so much of our data? And so it fed into the whole concern about that, the encryption that's going on, the Snowden disclosures. The various stories about this talked about how what this meant in terms of technology exactly fit what the Snowden slides had disclosed about the way the NSA was able to get into this. Now, the only problem is we talked about this 22 weeks ago on Episode 509. That was our "Imperfect Forward Secrecy" episode.

Leo: Right.

Steve: Based on a paper about how the Diffie-Hellman key agreement protocol fails in practice. And remember, this was tied to the Logjam attack. Again, good name, so it got lots of attention. Logjam, oh my god. And what Logjam was, it was again the sort of attack we've seen many times through the years, a so-called "downgrade attack," or an encryption-strength downgrade, where because many servers still support so-called "export grade" or deliberately weakened key lengths, it was possible for an attacker to intercept the handshake between a client and a server to lead the server to believe that the client only supported 512-bit Diffie-Hellman key agreement. And if the server did, then it would sort of shrug and go, well, okay, if that's the best you can do, fine.

The problem is that 512 bits is just no longer enough. And it's not that it wouldn't be enough if the prime numbers that were used were novel. But everyone uses the same one because in the protocol the prime can be known. The prime doesn't have to be secret. So no one bothers to create their own primes. They just reuse the same one. The glitch is, and this is what we discussed 22 weeks ago, is the way you break Diffie-Hellman allows you to do a lot of precomputation. That is, based on that prime, you can do almost all of the work, relative to the total amount of work you have to do, such that, if there's a communication link that is trying to come up based on 512-bit export grade, which is that is to say weakened, in this case it was a bank of PlayStations that they ran for a few days, and it cracked it. It did enough precomputation that, when a communication session was actually being set up using that non-novel prime, they were able to crack it on the fly and obtain the agreed-upon key and then decrypt the conversation.

So all of this we covered. Well, what happened was, I don't know, I guess they decided it was time to stir the press up again, so they highlighted - and this is all they did, 22 weeks later, was highlighted a different aspect of the paper that they had already

published, and that we had already covered. And that was that, while they with their PlayStation array - or maybe it was a GPU array. I don't remember now. But these guys, with a modest budget, were able to do the precomputation for 512-bit.

They announced, in the original paper, that a nation-state actor probably had the resources to do this for 1024-bit primes. That is, for primes twice as long as the export grade. Meaning that, while we fixed the Logjam problem, there was still the fact that most sites, or many sites, or, well, yeah, most, do still offer 1024-bit primes, and that somebody with lots of computing resources, like an NSA-grade operation, might have long ago done the precomputation because, once again, novel primes are not being used, even at 1024-bit length, because they really, you know, the agreement in the crypto community is that it's not that important. But that "not that important" assumes that precomputation is infeasible. These guys have shown, eh, so if you're big enough, you've got enough computing resources, you can do it. And the problem then is that the primes are not novel. So that's what all that was.

And to give you some sense of numbers, because they've analyzed the web, they said: "Breaking the single most common 1024-bit prime" - that is to say, looking out over the web at the Diffie-Hellman protocol, there is a most commonly reused 1024-bit prime. If that was the target of a nation-state scale precomputation attack, that would "allow passive eavesdropping on connections to 18% of the top one million HTTPS domains." So nearly one in five of the top one million could just be passively decrypted. And "A second prime would allow passive decryption of connections to two thirds of all VPN servers and a quarter of SSH servers." So, and then in their paper they say: "A close reading of published NSA leaks shows that the agency's attacks on VPNs are consistent with having achieved such a break."

So anyway, this is kind of old news, but still interesting. And there's a site, WeakDH.org. And I have a link in there that doesn't seem to be - I couldn't see it sitting on the home page. If you go to WeakDH.org, it has information about this. But if you go to - and do this, Leo. The link in the show notes, /sysadmin.html. That takes you to a page, which is not obvious from the home page, that lets you test anyone's server that you like. So, for example, put GRC.com in to the server test, and Leo's typing it now, and bingo. Good news.

Leo: Of course now I've got to do TWiT.tv. Hold on. Uh-oh.

Steve: Bad news.

Leo: We use a commonly shared 1024-bit Diffie-Hellman group, and it might be in the range being broken by a nation-state. Of course, we save no information on TWiT.tv about you, so I don't, you know.

Steve: Yeah, yeah. So anyway, again, WeakDH.org/sysadmin.html, for anyone who wants to poke their favorite website and see whether...

Leo: On the other hand, Bank of America does probably have some information about me that I wouldn't want a nation-state to own.

Steve: That's a little more of a concern, yes.

Leo: Whoopsies. Hmm.

Steve: Yeah. And so what that means is that it might very well be that a passive eavesdropper with sufficient resources could simply cut through the communications. And if anyone is running a server, the EFF's page that I link to is titled "How to Protect Yourself from NSA Attacks." And essentially it just means using Elliptic Curve Diffie-Hellman, the ECDHE, rather than the RSA-style, you know, based on primes, just standard DHE. So you can - and all clients are able to support that, and it's a pretty easy - it's a simple change to make to the TLS protocol suite in your server.

Leo: Well, I'll go talk to our server guys.

Steve: Yeah. So, Let's Encrypt. We've been following them for months. We began talking about them earlier this year, excited that they would be bringing this thing online. Initially they said the summer. And, you know, as with a project this big, it's dragged a little bit further out than they expected. But they've achieved another milestone. Last time we talked about them, they had used their protocol to sign the first cert.

So just to remind people what Let's Encrypt is, it is a joint effort by a bunch of big players. It is being hosted, the effort is, by Mozilla, Akamai, Cisco, the EFF, IdenTrust, the Internet Society, and sundry others. And the goal here is to automate the lowest grade of security certificate so that it literally removes that DV cert, the domain validation cert, from commerce, where it just isn't something that you need to say, oh, is it worth going secure or not? I'd really - I don't want to pay. I don't want to have to remind myself to, you know, what happens when I'm on vacation, and the certificate expires, then no one can connect to the site, blah blah blah. All of that goes away.

So Let's Encrypt becomes a CA, a certificate authority with an API, an over-the-Internet API, which allows the server to, on the fly, obtain a certificate for no cost. In interacting with this automated Let's Encrypt CA, it validates that it controls the domain. It then receives the certificate and brings it up and instantiates it on the website, and automatically renews. If the certificate does get loose, it's able to immediately revoke it and reissue itself a new one. Anyway, this is going to be a great step forward.

So the problem is they're a new CA. Our browsers all know about the existing CAs, but we don't know about Let's Encrypt. What they announced on the 19th of this month they achieved was that their own intermediate certificates have now been cross-signed by IdenTrust, which is a root CA that we all already trust. So what that means is that the Let's Encrypt-issued certs will be trusted by all clients everywhere. Over time, the Let's Encrypt root will end up being adopted by our clients. You know, Chrome and Windows and Apple and Firefox will put the Let's Encrypt root certificate in their root stores. Then the cross-signing won't be necessary. But this is, you know, this is a nice way to get bootstrapped and to immediately have servers able to issue a certificate which all clients will trust.

So we don't have any firm date yet on when this thing will actually go live publicly. But it's online, and sort of like I would say probably in late beta stage and really getting, I mean, this had to happen. Until the certs that they were issued were widely trusted, really it wouldn't be worth anything. So the fact that that has just happened on the 19th

says that, you know, this is very new, but moving forward. And I just - I think that's great. This is going to be - this changes the dynamics of the industry such that there just isn't a reason not to encrypt. And I imagine we'll see widespread support through all the OSes. I don't know when Microsoft will add it to IIS or to their server platform. I've seen no announcement one way or the other. But the various open source servers, Linux and Nginx and so forth, they're already there.

So we talked a couple weeks ago about the interesting concern about the 80-round hash collision which some researchers were able to create. And you know that one had a bank of PlayStations or GPUs, too.

Leo: Something, yeah.

Steve: So I don't know, I may be confusing myself because, you know, basically that's what you need these days is you need a wall of things that, if they're not doing that, they're bitcoin mining and making some money for themselves. But in this case they're busy trying to create a hash collision. So what this did was this freaked people out about SHA-1, more than they were already concerned. And as I mentioned at the time, you know, Schneier is the one who was quoted as guessing, just sort of just ballparking, as he did years ago, like what year it would be that the cost to crack had come down enough.

So, and you'll remember also that a week before the news of this very concerning, full 80-round collision was announced, the CAB Forum, which is the industry forum for certificate authorities, they floated a ballot to solicit votes from their members, all of their CA members, about allowing SHA-1 certs to continue to be issued in 2016. As it is now, no new SHA-1 cert will be issued past the end of 2015, past the end of this year. But at least one major entity, some sort of Fortune 500-style company, said that they had thousands of clients, I guess, or somethings, they had thousands of certs that they needed, that they would not have an opportunity to make the move in time. And no one on the outside got any more specifics than that.

So the question was, you know, in the wake of this breech, how would this ballot go? Was this going to change the trajectory of the voting to, in the wake of this, have these industry heavyweights at the CA Forum saying, uh, we don't think that's a good idea. So anyway, so I plowed into the mailing list, which is public, and followed the thread of discussion to see what they thought about this.

Geoff Keating, who's at Apple, he's the Apple representative in the CA Forum, said: "We've discussed this ballot," that is, this extending SHA-1 into 2016. They would all have to expire by the end of 2016; but the idea was, you know, so we're not letting them go any longer on the back end, but we're considering still allowing them to be issued at the beginning of the year. So Geoff says: "We've discussed this ballot within Apple. And based on what is known about SHA-1 security and the impact on an orderly industry-wide removal of SHA-1 support, we are against extension of certificate issuance until the end of 2016" - meaning keep it at the end of 2015 - "and so intend to vote against the ballot."

Erwann Abalea, who's at OpenTrust, he wrote into the list: "Was just reading it." And that is, he was referring to this new paper. "The complete 80-rounds SHA-1 compression function is broken. Some could argue that we still have a small security margin because of the choice of initialization vector, or the difference in work factor between collision and chosen prefix collision, et cetera." And that's what I was talking about when we discussed

this before. "But it took too many years to get rid of MD5," and he says, "(at least seven years after collisions were publicly demonstrated)." So, he says, "Let's do things better with SHA-1."

Then Rick Andrews, who is at Symantec and was the lead on the ballot, put into the mailing list: "Symantec and the endorsers withdraw this ballot." And then a guy, Gervase Markham over at Mozilla, chimed in, saying: "I'm not sad to see this ballot go." So that's it. No more SHA-1 will be issued after the end of this year. They can live through the end of 2016. But even Microsoft, who originally set that 2017 date, before the various browsers decided to make various warnings of their own ahead of that, it's over. So whatever company this was that said it can't possibly update itself in time, well, I have a feeling that their IT department will be very busy between now and the end of the year.

So, Leo, miscellaneous randomness.

Leo: Okay.

Steve: "Bridge of Spies" was fantastic.

Leo: Wow. One thing you and Dr. Mom agree on. We found it. I never thought that would happen.

Steve: Don't go any further.

Leo: Yes, everybody's seen - I haven't seen it yet. I'm dying to see it. This is the Spielberg film with Tom Hanks. And it's the story, I gather, I don't know how true, fairly true, I guess, I'll have to look into it, of Gary Powers in the U-2 spy plane; right?

Steve: Yeah. We don't know - yes. We don't know how much liberty the screenwriters took, although after the movie is over, before the credits begin, they do that nice thing where, for all the people that you've really gotten to know and care about, they give you a little paragraph on how their life went after that.

Leo: Gary Powers ended up, I don't know where he ended up, actually. I don't think it was anywhere good.

Steve: Reuniting with his wife and his family, who forgave him, and blah blah blah. So anyway, it was one of those where, you know, I was exhausted. It's like, what, two hours and something, two and a half or something. And the last 15 minutes you're just holding your breath. And Tom Hanks, you know, he's an actor for the ages. He's just like, you know, he does a great job. So for what it's worth, loved "Bridge of Spies."

And I did hear you mention, I think it was on TWiT on Sunday, that the "Steve Jobs" movie has pretty much fallen flat in the box office. I think it did, what, 7.8 million or something.

Leo: Yeah, yeah. Rightly so, yeah.

Steve: So certainly not a huge turnout. Oh, and I just got a kick out of this. I found this in the mailbag, so I thought I would share it without tying up one of our Q&A questions. A listener of ours in the U.K. wrote, the subject line that caught my eye was "Ling's Cars."

Leo: Oh, boy.

Steve: And he said...

Leo: Oh, we love Ling, yeah.

Steve: Yes. "On last week's episode I noticed you found LingsCars.com."

Leo: Yes, we did. Yes, we did.

Steve: "Ling," he writes, "is something of a minor celebrity here in the U.K. and is probably best known for using an old mobile ICBM missile launcher as an advertising billboard."

Leo: Oh, I love her. I just love her.

Steve: She's a real character.

Leo: She's a character.

Steve: And there's a link in the show notes for anyone who's interested. "The BBC," he writes, "did a piece on her some time ago." And then there's a link. I didn't ever make time to click on it, and I wasn't really that curious. But for anyone who wants more Ling: youtube.com/watch?t=672&v=cc1ktZRZ5ZM.

Leo: It was actually a "Dragons' Den," which is the original British version of "Shark Tank." She went in there to, I think, get money. Or maybe this is a piece about her that mentions the "Dragons' Den." Maybe that's it. Because it started off...

Steve: She is just a kick.

Leo: Yeah, no, this is - I feel like this is her "Dragons' Den" appearance. Yeah, she's

a character.

Steve: Yup, and there's the ICBM launcher.

Leo: The ICBM.

LING: My famous Chinese nuclear missile truck, my trademark. I'm looking for investments of 50,000 pounds for a 5% share of my company.

Leo: They didn't give it to her. And she's done so much better as a result. In fact, I emailed her, you know, we've had that little correspondence, Ling and me.

Steve: Good.

Leo: And I asked her to be on "Triangulation" because I'm dying to know more about her. She was trained as, I think, a chemist in China, a scientist of some kind.

Steve: A biochemist, yeah, I think.

Leo: Yeah. And obviously super smart. So she just did it all on her own, and she says, "Well, we do about 800,000 pounds a year." So they should have invested. I think it's a great story.

Steve: Whoopsie, yeah.

Leo: These guys are very not Ling-friendly.

LING: Yes.

ANNOUNCER: What on earth is with the advertising on the nuclear truck?

Steve: Oh, they don't get her at all, do they?

Leo: They do not get - they have no sense of humor. No sense of humor.

Steve: No, none.

Leo: Yeah.

Steve: It's like a dry Brit. It's like, come on.

Leo: Yeah. Yeah. And she's showing them the website, and they're just frowning.

Steve: I mean, and obviously it's she's just having fun. Who else, I mean...

LING: I have all these crazy ideas about how to market my business, and it works.

Leo: She's done very well, yeah.

Steve: And it works.

Leo: And it works. And that's the point, yeah. I love Ling.

Steve: Okay. You've got to get her on.

Leo: I've got a crush on her, to be honest with you.

Steve: It's just perfect.

Leo: She seems so cool. I guess this is more of a larger story, a piece of a larger story.

Steve: How long is that video? Can you see the...

Leo: Twelve minutes, yeah.

Steve: Oh, okay.

Leo: Yeah, it's not super long.

Steve: So anyone who is interested, the link is in the show notes for 12 minutes more of Ling.

Leo: Ling.

Steve: I also found in my mailbag - and I just thought I would do this because, why not? A belated birthday shout-out from Michael Cykowski, who's in Rochester Hills, Michigan.

He wrote: "Dear Steve, this email is about my father, Mark Cykowski. His 70th birthday is October 19th, 2015." So that was Monday before last, the day before our previous podcast. And of course I didn't see this until now because I didn't suck the mailbag down until this Q&A.

Michael says: "He has spent his life working in technology and hasn't missed an episode of Security Now! since 2008. I realize you aren't Ryan Seacrest, but is there anything I could do to convince you to give him a birthday shout-out on Security Now!? Could I donate to your favorite charity?"

And he says: "We have already purchased a copy of SpinRite, and it saved my butt twice. In any event, thank you for doing what you do. Your show is consistently one of the top tech podcasts out there. Sincerely, a big Security Now! fan on behalf of his security-obsessed father." So that was Michael writing. And to Mark, Happy 70th.

Leo: Aw. That's a big one.

Steve: Yeah. A decade away for me. Little more than that for you.

Leo: Not much more.

Steve: And I did find - this was interesting. This was on SlickDeals.net. I don't even - someone must have tweeted it to me, or I would have never known. But the URL is grc-steve-gibson-s-spinrite. And so the subject was GRC & Steve Gibson's SpinRite. Dale_101798 posted: "In the past I have recommended SpinRite to recover data from unresponsive hard drives. Some Slickdeals users complained that, on the huge hard drives we use today, SpinRite is simply too slow. However, if you want to recover data from any hard drive, you should at least know about SpinRite.

"Yesterday I found myself in the unenviable position of needing SpinRite to correct a failed hard drive in my wife's computer. It gave a blue screen error on every attempt to boot. Yes, we have backups; but restoring a backup on a new hard drive takes a long time, too. Got to go to the store, buy the hard drive, install the hard drive, and fire up Acronis True Image" - or Acronis? Acronis?

Leo: Acronis, yeah, I think you're right, yeah.

Steve: "...Acronis True Image and wait for it to do its magic. Instead, SpinRite did its magic on a 1TB hard drive in two hours, and I was once again the in-house miracle worker/computer genius. If you would like to learn more about SpinRite, go to GRC and view the video." So Dale, whoever you are, posting out in public, I thank you...

Leo: Thank you, Dale.

Steve: ...for sharing your experience with SpinRite.

Leo: All right, Steve. I hope you've got a cup of coffee or something, and we can launch into these questions, if you are ready.

Steve: Absolutely.

Leo: All right. I've got some good ones for you. People always want to talk to Steverino. These, by the way, come in from Steve's website, GRC.com/feedback, but also from the Twitter, @SGgrc.

Question 1 is talking about something you do on your website: Haystacks. Listener Jim says: Steve, love your Haystack interactive brute-force password search space calculator. That's there on your website, probably GRC.com/haystacks, I would guess.

Steve: Yup, yup.

Leo: He has guessable URLs, which I really like. Why is it, when I enter two numbers, like one and two, I get an exact search space size count of 110, and not 100? Aren't there only a hundred possible passwords for two-digit passwords? Steve, your math is off.

Steve: So I put this in here because the question comes up a lot. People, and I love the fact that people are paying that close attention, and I think they're just wanting to understand how it works. The reason is that, while there are 100 two-digit passwords using digits zero through nine, there are also 10 one-digit passwords.

Leo: Oh, you are so smart. What a smarty-pants. Oh, man. See, I would never have thought of that. This is Steve Gibson, ladies and gentlemen. Just right there in a nutshell. Wow.

Steve: So, yeah. Because this is the search space size, we have to search all passwords up to that size, using the alphabet which has been presented. So I did the math very carefully, and that explains the discrepancy.

Leo: Well, as long as we're correcting math, yours is correct. Mine is wrong. Richard Branson put $28 million into the Ring Video Doorbell, not 78, 28. Not that that makes much of a difference.

Steve: Still, yeah.

Leo: It's pretty good.

Steve: Definite vote of confidence.

Leo: Vote of confidence. Question 2 comes from Patrick in one of my favorite towns in the world, Laramie, Wyoming. Oh, Patrick's been hit by that interstitial advertising from Charter: Last week I started noticing what appears to be intrusive advertising on behalf of my ISP, which is the Charter Cable Co. Specifically, it seems like they're injecting HTML into websites. Here's a link to a screenshot I took displaying the advertisement: http://imgur.com/ncJENdi. Right there it says "VISIT THE NEW CHARTER.NET TODAY."

Websites appear as normal, but are shifted down by the height of this advertisement. So they're injecting it above, at the top of the page. In the photo, the color bands on the bottom of the image are the website's actual header, which would normally, of course, be at the top of the window, but because of this ad are not. Have you seen this before? What can I do? I'm going to add one additional thing. What does it mean? I mean, how are they doing that?

Steve: So, yeah, this was a great question. And it also made the cut because I'm afraid this is a sign of things to come.

Leo: More and more.

Steve: Yeah. Hopefully they're only able to do this with HTTP connections. That is, unsecured pages, and not HTTPS. I'm worried that there will come a day when your ISP will require you to accept a certificate from them, in much the same way corporate filtering firewalls do that for the machines within the corporation, so that those filtering TLS proxies are able to crack open the secure connections to check for malware, to check for content aspects and so forth. I don't know. I mean, this is going to be very controversial when that happens.

The good news is, in the same way that the Internet is going dark for law enforcement, and I don't mean that's good news for law enforcement, that's an inevitable consequence of encryption, but it means that it's going dark for this kind of conduct on an ISP's part because, if they don't force their customers to accept a certificate for their own proxy, then they can only do this on HTTP pages. And as we were talking about with Let's Encrypt, there will soon be no reason for any site not to be offering HTTPS connections.

And what's really annoying about this, this reminds me of the paper mail that I receive from Cox, my own cable modem supplier. I mean, they're an important piece of my infrastructure. They're the way this podcast is being delivered right now. I'm very happy with the service after that initial little transitional bump due to cabling when we switched away from the T1s. So I'm happy with everything. But they send me envelopes marked "Very Important Information Inside." And I dare not throw them away because maybe it is. But it isn't. It's trying to get me to use cable modem phone service. And that's the last thing I...

Leo: I hate it. Oh.

Steve: ...I ever want to do.

Leo: Now they've got a quadruple play. I don't - it's security, phone, it's everything.

Steve: Oh, and it's crap, I mean...

Leo: Of course it is.

Steve: Mark Thompson has it in Phoenix. And he's like, all of our conversations are, like, chopped up, where it's like his words are being chopped. And it's just like, no, no, no, no. Give me - I want to stay with copper. But so here we have "Important Action Required" coming up at the top of a web page.

Leo: Terrible.

Steve: And so it's like, what, what, what, what, what? And then "VISIT THE NEW CHARTER.NET TODAY." So this is advertising. And they're labeling it "important action required." And doing this to a web page that - basically changing the page that you're receiving. So it's breaking the, you could argue, breaking the copyright of the page that you're visiting so that you're not seeing what the website you're visiting intended you to, purely for their commercial purposes. So, wow. Yeah.

So the answer is it's certain that today it only happens over nonencrypted pages. So increasingly, sites are going full HTTPS. If you really want to get around this, a VPN will do it. The VPN will essentially mean that everything passing through Charter is encrypted. And so you could use proXPN, which is oftentimes a sponsor of the podcast and the TWiT network. Or whatever. But the idea being that encrypts your connection. Charter cannot penetrate it. Even in the future, if in some dark future customers were required to accept a certificate, you'd still be able to create a VPN connection, and then you get your pages without adulteration.

This is just, ooh, boy. I mean, it sort of comes and goes. We've talked about it before. We've seen it. But this is just, I mean, this thing is, what, like four inches, I mean, it dominates the page. If it was on an iPad, it would take up half of your iPad screen before you even got to the page that you were trying to visit. Ugh.

Leo: Terrible.

Steve: Really.

Leo: Anonymous Listener, my favorite, offered another great home filtering suggestion. You mentioned Pi Hole, but you can also set up a Pi, or any Linux box, with Privoxy as an adblocker. Sounds like maybe a hair growth treatment, but it's not. Privoxy, P-R-I-V-O-X-Y, is an HTTP proxy, so it's like privacy proxy. It's more work to set up, but it will do things like regular expression blocking and whitelisting on host names, and it will examine HTML contents of HTTP connections for suspect contents. HTTPS makes the second of decreasing interest, but it's a nice alternative

to messing with DNS. It's Privoxy.org.

Steve: Yes.

Leo: He says "Remember Proxomitron?" Is it the same people?

Steve: Well, no, that was actually me saying that. And I should have made that clear. Because all of us old-school, down-in-the-weeds geeks remember Proxomitron, which was this very techie, a Scott somebody, I can't quite remember his last name [Lemmon], was the - it was the brainchild of one guy who operated it for a long time. It was a proxy that you could run in your machine. And this is back in the day when almost nothing was HTTPS. You might briefly switch in to do a password. So this was early days, where a nonencrypted filtering proxy could be very effective. And that was what this anonymous listener meant when he said his second point was that it will examine HTML contents of HTTP connections for suspect contents, meaning that it's able to see into non-HTTPS.

But I wanted to bring this to our listeners' attention, for if there's anyone who just really wants power, because Privoxy.org, and the Privoxy proxy, is sort of that same thing. Proxomitron allowed you to rewrite your "hosts" header, rewrite the "referer" header, I mean, it was a very powerful, almost a programming language for the browser interactions. And Privoxy is that scale of strength.

One of the limitations, for example, of the Pi Hole offering, where it's about blackholing DNS domains, is that it is limited. That is, you have to explicitly fully name any domain that's going to be blackholed. But, for example, with Privoxy, you could say, if the domain name has advertising anywhere in the string, using a regular expression, blank it, blackhole it, and so forth. So you could easily create a set of sort of general rules to apply, and then of course you could still do explicit matching on domain names. So anyway, thank you, Anonymous Listener.

Well, what happened was we have a few of these anonymous things because, when I pull the mail down, the mail is sorted into two folders. GRC's website at GRC.com/feedback is a form, and people are invited to put their name and location in because it's fun for us to - it just sort of makes it more conversational to say, hey, you know, Christian Steinway in Dallas, who happens to be the next question, so I see him there, has this question. As opposed to just sort of being anonymous.

So anyway, they both come down. And in this case I just sort of looked in the anonymous folder because I didn't want to, like, rule out people who didn't provide their name because I want to allow people not to have to do that. And a couple subject lines caught my eye, and so they made it onto this week's Q&A. Normally, though, I just go into the named folder, which is about 20 times larger in count. Which is to say, only one out of every four or so - no, 20 times larger, so almost everyone does get a shout-out that way.

Leo: Good. Question 4 from Christian Steinway. See, we know his name. We know where you live, too, Christian, in Dallas, Texas. He wonders about SQRL and "what you know": You mentioned recently on the podcast a legal distinction - actually this was me, so I won't put this on you in case I got it wrong - between it being legal to

be coerced to produce "what you have" (fingerprints, dongles, DNA, hair samples, that kind of thing) to access encrypted information, but there being strong constitutional protections against testimonial self-incrimination preventing law enforcement from coercing anyone to produce "what you know," like a password, for instance. It occurs to me that the use of SQRL would fall in the former category. Is that so?

Steve: So that's sort of an interesting question. So Christian was asking whether SQRL would constitute something you have or something you know.

Leo: That's a great question.

Steve: Which is really a great question because he understands that in fact what SQRL is, essentially, we could think of it as a secure proxy for you. That is, it is able to authenticate your identity across the entire Internet securely, in a way that cannot be tracked, and so that you don't have to know any of that crypto that it has with all of the different sites you visit. And this has been a subject of some controversy in the newsgroup while we were developing this because some people object to my insistence that the user type a password for every single authentication, my argument being that, if you walk away from your machine, anybody can go to your machine, go to a website, and use your SQRL to authenticate as you and log in as you.

So I have always been lobbying for maintaining a "something you know" component. And the idea being that, while, yes, technically it doesn't need that at all, you know, that password is simply unlocking its ability in a secure way so that without the password it's very, very difficult to unlock. Still, I think that's an important interlock.

But the other concept to remember is that that unlocking part, that password part, is really not part of SQRL. SQRL is this trans-Internet authentication system. The password is sort of an implementation detail. And by that I mean that, now that we have access, apps have access to like the fingerprint scanner in iOS, Jeff in the U.K., who's been developing the SQRL client, we've had thumbprint support for months so that you just put your thumb on the button, and you're authenticated.

Now, the danger, of course, is now we're back to something you have. You have a fingerprint. And courts have ruled that you can be forced to divulge your fingerprint. But the nice thing is this puts the user in control. And so as long as you know what the law is, then you have an authentication system. And of course SQRL's not unique. Anything that's able to do this, like a dongle that requires something more than just its own presence, it falls into the same category. So, great question.

Leo: I love it.

Steve: And I guess the short version is it's up to the user. We can support either. Not requiring something you know or adding that to the mix for extra security.

Leo: Yeah, for two-factor, which we like.

Steve: Yup.

Leo: Vosguard asks about the current status of VeraCrypt. This is a product that many of our users have recommended and used - you have not - as an alternative to TrueCrypt: Steve, you said it was time to update TrueCrypt, but then three weeks ago you said that VeraCrypt was not ready until they did a update. There's a flaw that we were talking about.

Steve: Yup.

Leo: So should we move from TrueCrypt to VeraCrypt yet? I really count on you. Please advise. I'm still using TrueCrypt, and if I should move to VeraCrypt now, I need to know. Vosguard.

Steve: Okay. So it has been about a month and a half since we discovered a problem with TrueCrypt. The guys that are doing VeraCrypt - oh, and by the way, this was Google who, looking at the TrueCrypt source, realized there was a privilege escalation or elevation vulnerability such that it would be possible for a process running in your machine to leverage the TrueCrypt driver in a way that it wasn't intended to be, in order for it to obtain the same privilege as that driver, which is to say, root-level, full-system privilege.

So the VeraCrypt guys immediately fixed the bug, which was in their code, too, because they forked it from TrueCrypt. As we know, TrueCrypt will never be fixed. TrueCrypt is at 7.1a now and for the rest of time. So because of that, I recommended that, after 16 months, which is how long it's been since TrueCrypt went silent, went dead, it was arguably now time to move.

The bad news was that it quickly came to light that there was a bug in VeraCrypt's implementation of the fix. So the fix, the first fix, was v1.15, which they released on September 26th, so just about this time a month ago. They quickly - it quickly came to light that there was a problem. And our listeners will remember, the problem was with deleting folders on the updated VeraCrypt. You could delete files, but not folders. That caused some people to say, whoa, that's important for my use. I'm going to hold off.

About two weeks later, on October 7th, which is three weeks ago, they released 1.16 that fixes that. Now it's been three weeks. I checked. There's no obvious known problems. So, yes, Vosguard, and anybody else who was waiting, I think now it's safe, and probably worth migrating. Again, this isn't - at no point is your encrypted data at risk. This is just a way that, if malware got in your machine that had your hopefully non-admin privileges, and you had TrueCrypt or the older VeraCrypt, like before 1.15, installed, that malware could leverage this subtle bug in the original TrueCrypt driver to obtain higher access rights, admin-level access for itself. Again, this really doesn't involve the encryption, it just is a trick of the way that the driver was written. That's fixed in VeraCrypt as of 1.16.

So, you know, when it's convenient, it's probably worth migrating. And the migration is pretty simple. VeraCrypt will be able to mount non-system drives. If you have a system drive, that is, like your main C: drive, the boot drive is TrueCrypt encrypted, you'll have to first remove the TrueCrypt encryption from the entire drive, which as we know takes a while, then install VeraCrypt and reencrypt the system drive. But if your use is of

mountable volumes, then VeraCrypt is able to mount existing TrueCrypt volumes and is able to upgrade them to its own.

Leo: Very nice. So it is - so you say okay. Time to move.

Steve: Yup. Time to do it. It's not an emergency. It's not a panic. But anyone setting up a new system should use VeraCrypt. And if there's like a - if your use of the system potentially exposes it to this, then it's worth doing. My sense is that, in time, malware may start looking for old, non-upgraded TrueCrypt drivers and just use that, sort of add that to its bag of tricks.

Leo: Right.

Steve: And so you'd like to be out of the line of fire by the time malware, just sort of as one of the things it does, checks to see if it might be able to use that leverage.

Leo: Let's see. This is from Henry Adams in Ellicott City, Maryland. He's wondering about DNS in a double NAT setup: I've learned a lot about computer security from listening to the show. I can then turn around and keep everyone else in our big extended family safe on the Internet. I think that's one of the real good reasons to listen to this show. He says: I've read your article on "Multi-NAT Router Networks," and I set up my home network exactly that way. Is this that triangle thing, the three-router thing you were talking about?

Steve: Either two or three, where you have a network inside a network, yes.

Leo: Right, right. I ran your Domain Name Speed Benchmark - this guy really loves you, Steve. I ran your Domain Name Speed Benchmark utility the other day. It gave me suggestions for faster DNS lookups. My question is whether these suggestions are applicable in a double NAT setup? Also, for the internal router, which DNS server should I point it to for the best performance? This actually is a great question. Should I pass it on to the external router, or point it directly to Google or OpenDNS or whatever's fastest?

Steve: So, yes, great question. And what you probably want to do is override the default, sort of as Henry is suspecting. When you have a router inside of another router, the internal router is making a DHCP, dynamic host configuration protocol, query to the outer router, the router that's plugged onto the Internet, for its information. And many routers now give their own gateway IP as the DNS IP. So rather than passing through the public DNS that external router has received, the router says, oh, no, just use me for DNS. Now, the inner router's probably going to do the same thing.

But so Henry's question is, first of all, that it looks like the DNS Benchmark said maybe use Google or OpenDNS. So it does make the most sense to manually configure that DNS on both routers. It will be faster if the internal router makes its request directly to the remote public DNS server, rather than bouncing it through the external router. There's just no reason to do that. There's no benefit. There's no value added. You're just giving

that external router a little more work to do, and there's no reason to. They're already - they tend to be sort of underpowered processors.

So, you know, shooting that packet just right through it, rather than asking it - and, see, and it's a little - there is some overhead because you ask it, and then it has to pend your query while it asks the public DNS routers. Then when it gets the answer, it responds to your query. So there's some overhead associated with that. Much better just to have the DNS query packet shoot right through the external router to the public DNS server and come back.

Leo: That's kind of like, well, see, isn't there overhead in general with double NATing?

Steve: Yeah. I don't think it's significant.

Leo: Okay.

Steve: But, yeah, but a little bit. But probably it's nice to have the isolation that another layer offers.

Leo: Yeah, yeah. Yet another Anonymous Listener gets a new IP address every time he connects. We were talking about I think DynDNS and the issue of dynamic IP addresses versus static IP addresses and that kind of thing. Perhaps most people's home router IP address doesn't change often, but my Internet provider, a Canadian company, AEI.ca, changes it every time I connect. Hmm. So it does depend on your provider what kind of lease expiry is in force. Thus it is a local phenomenon.

Steve: So it's more than that. And so many - it's interesting to me how many people chipped in with their own experience.

Leo: Because I was saying never, mine never - Comcast never changes my IP address. [Crosstalk].

Steve: Right, and I had mentioned that last week, when I woke up, I saw that there had been a block of about an hour around 1:30 in the morning that I was just off the 'Net. And then I came back up, and my router reestablished my static OpenVPN connection to GRC. Everything was fine, but my IP was changed.

Leo: Hmm, interesting.

Steve: So I wanted to broaden this a little bit and say it's not so much a local phenomenon as much as there's absolutely zero obligation on the part of the ISP to keep their subscribers' IPs static. That is, it's just sort of a, you know, the reason they're not changing is there would be some brief interruption of service, potentially, if it changed, and they don't want to do that. And there's typically no reason to change them.

But maybe, for example, that hour outage I had a week ago, where my IP did, actually it jumped a fair distance, that may have represented some major replumbing that Cox was doing. Maybe, for example, a new housing project was opening, and they needed to move a bunch of IPs from one network leg to another, you know, who knows. It's all sort of mysterious and behind the ISP. But my point is that the user's public IP just doesn't matter. As long as it's not changing constantly, as long as it's relatively static, and, for example, maybe make the change at 1:30 in the morning when everyone's asleep, then there just isn't any obligation.

So some people use sort of the old PPTP, the dialup-style DSL. That's what they connect to the Internet. So, yes, and their router does that every time they connect. It's very likely going to pull an IP from a pool and get a different one. Other people have a cable modem that's statically on with a router behind it and a lease that is, for example, 24 hours. And when the router says, hey, my lease is expiring, give me another IP, DHCP allows the router to tell what its current IP is so that, unless there's any reason for it to change, the ISP typically says, oh, you just can use that for another day. And, you know, call me back in 24 hours, and we'll talk again.

So anyway, the point is that, yes, there's like, it's all over the map, that is, the behavior of how static IPs are, because it's purely a function of convenience for the ISP, and we're sort of lucky just to have an Internet connection.

Leo: We're just lucky, that's all.

Steve: Eh, just lucky.

Leo: Lucky we got an IP. A listener wonders about NoScript with uBlock Origin. I have some tails to tell about uBlock Origin.

Steve: I do, too.

Leo: Yes. Steve, regular listener of the SN podcast here. What's your thinking regarding NoScript and uBlock Origin? Have you stopped using NoScript in favor of uBlock Origin? Thanks.

Steve: So I mentioned this a little bit last week, but I wanted to make sure it was received. And that is, yes, not only did I - first I flipped it off and ran with it that way for a few weeks. Then I kept looking at that icon sitting there in my Firefox toolbar, thinking, eh, and I removed it. That is, I removed NoScript. And for me, and I think maybe we're back to the frog in the pan of cold water analogy again, where you turn up the heat slowly, and you cook the frog, whereas if you toss the frog in an already hot pot of water, it jumps out. What happened with me, since I've been using NoScript for years, is the pain threshold, I'd just become adapted to, like, things not working. And it's like, okay, you know, enable scripting and then do something. And, like, buying things, or any sort of a complicated negotiation with a website, it just wouldn't work. And so what I realized was, boy, you know, it's sort of like you're just holding your breath and clenching. Or sort of like that noise that you weren't aware that was going on, and then when it stopped it's like, oh.

Leo: Thank god.

Steve: Thank god it's over. But you weren't even aware of it. So that's how I feel. I have scripting back. Now, yes, we've been talking about the dangers of scripting. But most of the problems these days are coming from add-ons which are under control, like click-to-play with Flash. Java has pretty much moved off of this reservation. And ads are now being controlled by uBlock Origin. So for me, and I get it, people may still want to be behind NoScript. I wouldn't fault them. But, boy, it's just every time I do something, and it just works, I think, wow. That's the way it's supposed to be. Rather than what I'd, you know, the corner I'd painted myself into.

So, interested to know what you think about uBlock Origin, Leo. I've been intending to contact the author because a site that I use frequently, iHerb.com, absolutely will not work. Even whitelisting it. It must be that they're using other domains from the main iHerb.com domain. And then this is on my iPad. Just I have to go turn it off, or go to a different browser. And it just - and that's wrong. There ought to be a way to say "whitelist this domain sticky for everything this domain does." Or maybe "whitelist this tab until I close the tab." So other domains that derive from that root domain also get whitelisted. But, you know, this is a real problem, if you've got - especially for the target audience for these filters, which are people who are not tuned up and techie, to like have it completely break a site so that nothing you do, even whitelisting it won't fix it.

Leo: I haven't had anything that bad happen. But I just - there are some sites I can't get to, like SourceForge, because he's decided - or one of the blacklists he uses.

Steve: Oh, yeah, yeah.

Leo: Yeah.

Steve: Yeah, SourceForge, for example, yeah. It comes up, and you're able to say "allow forever" or "allow temporarily."

Leo: Yeah. And that's what I should do, although I'm not a fan of SourceForge, so I can...

Steve: Right, so it's like, oh, good, you saved me from going any further.

Leo: Right, right. But then also it blocks websites - so a lot of times when you search for a website, you get an ad linked to the website, in addition to the search result; right? Let's see if I can make this happen. If I click the ad link, which is usually the top link, uBlock will block it, even though - I guess because it's going through Google's server. And it blocks it in such a way that it's really ugly. And then, but then if I go to the next one down, it's fine. So it's kind of - it's minor. That's minor compared to not being able to ever unblock a site.

Steve: Oh [laughing]. Yeah, in fact I'm sure I can get to it on Firefox, and I've got - oh, of course, no, wait. I was thinking uBlock - okay. He was saying uBlock Origin in Firefox. I was referring to uBlock. But actually I was referring to...

Leo: NoScript and uBlock Origin. No, that's the one we use, uBlock Origin.

Steve: Yeah, but what about on the iPad?

Leo: Oh, on the iPad you've got Crystal and, what was it...

Steve: Exactly. Is it rBlock?

Leo: No, no. I have it here. I have it on my phone.

Steve: As do I, but I don't have a iOS device...

Leo: Yeah, 1Blocker.

Steve: 1Blocker. That's right. 1Blocker. And so, I'm sorry, so it's not uBlock Origin, it's 1Blocker.

Leo: Oh, 1Blocker, oh, okay.

Steve: Yes. So on your phone right now try iHerb.com, I-H-E-R-B.

Leo: What are you buying at iHerb?

Steve: It's supplements.

Leo: Oh, okay.

Steve: It's like the goto place for supplements.

Leo: Okay, iHerb.com. Vitamins, supplements, and natural health products. Sounds great. And the result I get on my iPhone is a site.

Steve: Oh. I don't think it even comes up for me. Or maybe if I...

Leo: No, I think I'm using, wait, let me see which one I'm using because...

Steve: Oh, yeah, right.

Leo: I'm pretty sure I'm using 1Blocker, but let's go to Safari. And...

Steve: Content filters.

Leo: Content blockers, yeah, 1Blocker. And it's on.

Steve: Huh, interesting.

Leo: So that's really weird.

Steve: I haven't tried it on my phone. Maybe the phone and the pad implementations are different.

Leo: That's very odd.

Steve: Because, you know, the phone screen still needs to be bigger.

Leo: Well, you know, where you really, I have to say, where you really need an adblocker is on mobile because a lot of these sites just are terrible.

Steve: Yeah, yeah.

Leo: I got the mobile site. They want you to get an app, but who cares about that. Get the app, you won't have to worry about it.

Steve: Right. Although the app is iPhone only, so that's annoying, too, because, I mean, I really - my iPad is my goto device. I spend more time with it than - or iPads, because of course the car has its own, and the bathroom has its own, and the bedroom has its own, and the living room has its own.

Leo: Wow, wow. You're amazing.

Steve: Eh, well.

Leo: Moving along to Number 9, Brennan in...

Steve: They're all plugged in, charging all the time.

Leo: They're all charging all the time.

Steve: All topped off.

Leo: Always ready to go.

Steve: Okay, Brennan.

Leo: You crack me up. Brennan in Vancouver, Canada brings news of a free computer science degree for all: I think some listeners of Security Now! would be interested to know about a free computer science degree put together by a group called the Open Source Society. The group has created an index of computer science courses, many of which are hosted by some of the best universities in the world - Stanford, Harvard, MIT, UC Berkeley. All courses are free and can be taken entirely online. In order to show that you've successfully completed a course, you create a startup project where you solve a real-world problem using knowledge acquired from the course.

The index to the courses is available on GitHub at the link I'll give you in a second. At a time when university graduates are drowning in record levels of debt, alternative methods to education are becoming ever more appealing. And I don't know if it's an accredited degree, which may or may not make a difference to you.

[github.com/open-source-society/computer-science]

Steve: It is incredible, Leo.

Leo: Wow. GitHub.com/open-source-society, and then click the Computer Science link.

Steve: And scroll down a bit and look at the - so there's a big OSS, Open Source Society University. And then scroll down, look at this curriculum that they offer.

Leo: You have to be able to use GitHub to do it, which cracks me up. But of course, you shouldn't - and then, wow. Okay. The MIT challenge. The entire four-year MIT curriculum in one year. Introduction to Computer Science. Math (Mathematical Thinking). Program Design. Math (Discrete Math). Algorithms. Programming Paradigms. Software Testing. Software Architecture Theory. Wow, there's quite a bit.

Now, oh, here's the Introduction to Computer Science. It's a 12-week course, 10 to 20 hours a week. There's one using Python. And from NAND to Tetris. I like that. So these are from accredited universities. I just don't know if the - oh, this is on Coursera, this one. I don't know if, see, yeah, what you get is a certificate. That wouldn't be the same as an actual diploma. That may not matter to you, but just to be aware of.

Steve: Yeah. Looks like great education.

Leo: It's just aggregated all the different kinds of courses out there. Yeah, I think this is just, I mean, this is [crosstalk].

Steve: I think it's the future. I mean, this is, you know, it's crazy that we're still largely doing education the way we were pre-Internet.

Leo: Yeah. And what is the role of the university when you can do on-demand learning as needed? I think there is a role, but that's a bigger conversation.

Steve: Right.

Leo: But this is...

Steve: And I agree with you, by the way.

Leo: And you learn anything you want. I mean, that's great. Yeah, it's not - this is just one of many, many sources for this kind of stuff. iTunes University has many of the same courses. There's just a lot of places you can go now. If you're a motivated kid, and you don't have the means or the resources, but you have the desire and the drive, the sky's the limit.

Steve: Yeah.

Leo: Yeah, I agree. Nice, it's a nice time to be alive. Although, you know, the latest thing is that MOOCs are a flop, this massively online courseware? The reason is MOOCs are a flop is because so many, millions of people, you know, some of these courses have 30,000 people in them. And like Udacity courses. And that means 30,000 start, but very few ever finish. And this is the - but that's not their fault. This is how we are.

Steve: Right.

Leo: That's why I said, if you're motivated...

Steve: It's like, eh, you do it for a couple hours, and then...

Leo: Yeah, yeah, yeah. I can't tell you how many programming books I've started. It's about 10 times the number I've finished.

"TL" in Kentucky helps Steve correct a misconception. Our last question: I read your transcripts every week, and I see a theme in some advice you give which is only partially correct.

Steve: Whoops.

Leo: When Flash updates come out you constantly, constantly remind listeners they only really need to take action if they use Firefox, since IE and Chrome automatically update the Flash Player when updates are available. But that is only partially true, my friend. IE only updates Flash Player if you have Windows 8/8.1 or Windows 10. Oh, yes, that's true. If you have Windows 7, no matter what version of IE you have, even IE11, the Flash Player is still an add-on you have to keep up to date yourself. Chrome, being in its own world, does update regardless of OS version. We don't tell anybody use IE, do we?

Steve: No. But I do mention that, if you have IE, then IE takes care of Flash. And I didn't put an exception in for Windows 7. So TL, I thank you. I stand corrected. And I will be more careful in the future.

Leo: And if you're using Windows XP, and you're using Flash, you're insane.

Steve: It's true.

Leo: Stop it. All right, Stevie. We've gone through 10 great questions. You've answered them like a champ.

Steve: We're caught up on the news, and next week we're going to plow into the architecture of Objective C's method binding technology.

Leo: Neat.

Steve: Neat, neat, stem-winding propellerhead.

Leo: Fascinating stuff. This is my new Apple Watch. I just got sent this. I'm very

excited.

Steve: What?

Leo: No, I'm just kidding.

Steve: That looks like a Tamagotchi.

Leo: It does, doesn't it. I have no idea what's going to happen when I turn it on. Some fan sent this. All right, Steve. This concludes this Security Now! for this Halloween week. Will you be trick-or-treating on Saturday night?

Steve: I turn the lights out and hide. And actually it's been the case now, I'm in a community that was originally zoned as a retirement community.

Leo: No kids.

Steve: And there's like a retirement center not far away. But then there was some legal problem where it was illegal to actually say only old people on welfare could move or something. I don't know. So it's not that.

Leo: Get off of my lawn.

Steve: But still there just aren't any kids. And the new trend seems to be that everyone, the kids go to, like, big shopping malls, and that's where they do their trick or treating.

Leo: Yeah, parties. Yeah, or they go in a group to a neighborhood that is more felicitous to the young trick or treaters than yours.

Steve: Yes, a bunch of crotchety old people...

Leo: What do you want?

Steve: ...who are hiding in the house with the lights off.

Leo: I have my Ring Doorbell, and you go away. That's funny. Well, good. So you will manage to avoid this entirely. I, on the other hand, will be actually going around with a 12 year old.

Steve: You're deep in the middle of it, my friend.

Leo: He and I will both be riding Segways, though. So this is...

Steve: I was just going to go there. I was going to say, "Get on your Segway."

Leo: Yeah.

Steve: Yup.

Leo: This is going to be the best Halloween ever. And in fact my costume is Paul Blart Mall Cop, which is a movie you never saw, of course, because it was a terrible movie.

Steve: That is completely correct. I can't even pronounce that.

Leo: But it's a guy, a very funny comedian, oh, I can't - Kevin James, who looks, you know, he's a heavyset guy like me. And he's a mall cop who rides a Segway. And it turns out, not only do I have a physical resemblance, I have a Segway.

Steve: Oh, my lord. I see. So of course it fits right in. For anyone who has seen the movie, they'll know what you are for Halloween.

Leo: You'll know anyway because it's going to say "Mall Cop" on my Segway.

Steve: Perfect.

Leo: And I've got a hat that says "Security."

Steve: Perfect.

Leo: And I've got a police shirt from the Chicopee Police Department with a badge.

Steve: And a CB-style microphone.

Leo: And I've got a CB-style microphone I'm going to glue to this thing. And we've got a perp over here. I'll probably get shot.

Steve: Looks like we've got somebody doing tricks instead of treats over here.

Leo: I'm probably going to - I'm going to get beat up, for sure. "Hey, Mall Cop, come here. I got something for you." We can do this show every Wednesday until that happens. And I hope you will come back - Tuesday, Tuesday, Tuesday. Come back next Tuesday at 1:30 Pacific, 4:30 Eastern, 20:30 UTC. That's when we do the show live. You can always get it on-demand after the fact, TWiT.tv/sn, or YouTube.com/securitynow. Unless we get kicked off YouTube tomorrow because you know they're doing this new ad-free thing.

Steve: Something about Red. What's that about?

Leo: Yeah, YouTube Red. I don't know what that's going to - people tell me it's not going to affect us. But don't worry, Leo. It's not going to affect you. Eh. So probably still there, YouTube.com/securitynow. And of course on your favorite podcatcher. Good news, Google just announced that Google Music, which is on all Android devices, will now feature podcasts or soon feature podcasts. And we are a launch partner with them, so all of the shows, including Security Now!, will also be available through your Google Music. So there'll be lots of ways you can listen.

Steve: You'll have a hard time not listening, in fact.

Leo: You'd better listen.

Steve: It's coming at you from every direction. You can't get away.

Leo: Don't make me pull this podcast over. Steve is at GRC.com. He's got, you know, everything there - Perfect Paper Passwords and SQRL and of course this show.

Steve: Haystacks, yup.

Leo: Haystacks, Password Haystacks. This show is at GRC.com/securitynow. There's audio there. There's transcripts. He's got - he's the unique holder of the transcripts because he pays to have them done, which is awfully nice of him and Elaine Farris, who writes these. So you'll find that there, and SpinRite, the world's best hard drive recovery and maintenance utility.

Steve: Yup, keeps the lights on, pays the bills, and lets me keep coming back for more of this.

Leo: I think Paul Blart Mall Cop would wear an Apple Watch.

Steve: Oh, I actually think he would.

Leo: Just like that.

Steve: The question is, does a worm crawl out of that apple when you start that?

Leo: This is the pinkest Apple watch I've ever seen. I don't - oh, it's got a big plug. DC, 5 volts. Oh, geez Louise. I don't know. What is this? Recommendation: Fashion strap securely but comfortably around child's wrist. Do not over-tighten. Okay. I guess this is my first Apple Watch. Thanks, Steve. We'll see you next week.

Steve: Thanks, my friend. See you then.