SECURITY NOW!

Transcript of Episode #316

Listener Feedback #125

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

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-316.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-316-lq.mp3

Leo Laporte: This is Security Now! with Steve Gibson, Episode 316, recorded August 31st, 2011: Your questions, Steve's answers, #125.

It's time for Security Now!, the show that protects you online. Yes, a show can do that. Well, it can if it's hosted by this fellow right here, Mr. Steve Gibson of GRC.com, the man who discovered, coined the term, and created the first antispyware program. He's been so busy with his new passwords and encryption technology. He's been teaching us how to use the Internet and what the Internet does, how it works, on and on and on. But it's time for our semi-monthly Q&A show, Episode 125, Q&A 125.

Steve Gibson: It occurs to me, Leo, that anyone who's hearing this from a recording will know the answer to this question. But I didn't ask you, so I just wanted to make sure that we are recording this?

Leo: Yes [laughing]. No, it's not an inappropriate question.

Steve: I just normally check in, just to make sure, not that the reels are spinning, but somewhere there's a hard drive whose head is thrashing around.

Leo: Have I ever - have we ever forgotten to record this show? Oh, yeah, we did it twice once, didn't we.

Steve: We've only had to redo a couple in the 316 shows. So I'm not saying it's likely.

Clearly anyone listening to a recording already has the answer to the question.

Leo: I am no longer alone here, as you can see in our shot of the TWiT Brick House. People are working hard, running around like crazy people, making sure that everything gets done. Well, they actually look like they're fast asleep, actually.

Steve: You know, when I was there last week I didn't see a lot of big red Record buttons. I would have felt a little more comfortable if there were some big red Record buttons.

Leo: Yeah, it's funny. I think if you walked in here, you know, just completely untutored...

Steve: It was very quiet.

Leo: ...you would not know, A, that anything was going on; and, B, you would not know how to make anything go on. I used to know how to do everything here. And I no longer know anything about how to do anything.

Steve: That spinning throne looks like the bridge of the Enterprise. I mean, it's phenomenal.

Leo: That's our...

Steve: That master control unit.

Leo: Yeah, we've looked for a good name for that. Have we decided, John, on a name for that yet? I know that the person at the helm is called the technical director. And what do we call that master control suite? It's the turret. It's the control column.

JOHN: If you include the stuff in the basement.

Leo: Oh, yeah, you have to include the stuff in the basement because really the thing up above, the thing that rotates...

Steve: It's like an iceberg. You just see the little tip of it...

Leo: It's just the tip, yeah.

Steve: ...above the water.

Leo: In fact, it's really basically control surfaces and display screens that really go down to the basement where all the work is being done.

Steve: Oh, the basement is amazing. I just love the basement.

Leo: Yeah, yeah, it's fun. I have to say I'm really happy that we built this. It's just so cool. We're still, you know, we're still tweaking it. My office is the last to get kind of tweaked. You can see there's still wires lying around and stuff like that. But I just - I'm really happy here. I enjoy coming to work here. It is kind of peaceful in an unusual way, I guess because before we were just all jammed into one room together. We're still working on the kinks. We're getting the kinks out. But I'm pretty happy about that. So we have a lot of security news today. I know we've got questions and answers, too. So let's get down to it. Let's get right to it.

Steve: Get right in. So under Updates, which I always do first, technically everything got updated that we care about: Firefox, Microsoft, and Chrome. And presumably there's something from Apple, although I haven't actually seen the notice yet. But it's all because of what is top of our Security News, that all of the manufacturers and publishers of browsers have been scrambling around as fast as they could to deal with something that's sort of related to the Comodo problem that we talked about before.

This is a Dutch SSL Certificate Authority called DigiNotar, who for reasons that are still not quite clear - and I'm getting the sense that there's a little more to this story than is public yet, from looking at the source code to what Mozilla has done. It sort of leaks some information that we'll talk about. But what happened was a *.google.com certificate was found in the wild that had been signed by this relatively obscure - I guess they're not obscure if you're Dutch, but they're obscure for us, and certainly for Google - certificate authority DigiNotar.

Well, DigiNotar has a collection of root certificates in all of our browsers. There's a DigiNotar Root CA, a DigiNotar Cyber CA, a DigiNotar Services 1024 CA. Those are root certificates across the board in Mozilla, in IE, in Chrome, in Safari, in the mobile platforms. It's one of the collection of some 600 certificate authorities, any of whom have the ability to sign any certificate, which is part of the reason that people are beginning to feel more and more that this whole SSL/TLS trust model, this certificate authority anchor trust model, is becoming a bit rickety.

And our long-time listeners will remember, probably with a grin, the show, the podcast you and I did, Leo, where I first in a decade looked at the list of certificate authorities in whatever browser I was using, and I was stunned by the explosion of them. I remember back in the day there were seven. And now, I mean, and of course we've had a lot of fun at the expense of the Hong Kong Post Office. But they're one of the people who are able to sign any certificate that they want to on the Internet, and all web browsers will trust it without asking any questions.

Leo: So it's not the Hong Kong Post Office anymore, it's the Dutch Post, or whatever this is, Dutch Post Office.

Steve: And I guess it's sort of like a digital notary. So Microsoft posted their Microsoft little announcement, sort of in the standard Microsoft style, says: "Microsoft is aware of at least one fraudulent digital certificate issued by DigiNotar, a certification authority present in the Trusted Root Certification Authorities Store" - meaning in Windows - "on all supported releases of Microsoft Windows. Although this is not a vulnerability in a Microsoft product, Microsoft is taking action to protect customers.

"Microsoft has been able to confirm that one digital certificate affects all" - and this is a typo because I copied and pasted it. Anyway, they should have said "affecting all subdomains" - I guess it's right as it is - "affects all subdomains of google.com" because the certificate that was signed was *.google.com.

Leo: Oof.

Steve: Yeah, "...and may be used to spoof content, perform phishing attacks, or perform man-in-the-middle attacks against all web browser users including users of Internet Explorer. Microsoft is continuing to investigate how many more certificates have been fraudulently issued. As a precautionary measure, Microsoft has removed the DigiNotar root certificate from the Microsoft Certificate Trust List."

Leo: So anything that DigiNotar signs, not just ...

Steve: Well, yes.

Leo: But that's appropriate.

Steve: This is a, yes, well, this is how badly you get spanked these days if you're a certificate authority who screws up and through whatever means gets tricked into or...

Leo: You think that's what happened, that some malware author tricked him into this.

Steve: No one knows. There were some early unverified reports that Iran was using this certificate at their borders in order to spy on Iranian transiting traffic. But there was never any foundation for that. So, as our listeners know, I'm really slow to buy into these things. I mean, I was slow to buy into the fact that Stuxnet was targeted at the Iran nuclear program until it became very clear that we had the evidence to say that because it's just irresponsible to go off half-cocked.

Leo: So Firefox, Explorer, and Chrome have been updated. The chatroom's pointing out, quite notably, what about Safari?

Steve: I know. Now, I think I recall that Safari on Windows uses the Windows certificate store. I don't think Safari has its own, that is, the Safari on Windows. So Microsoft's removing it from the Windows trusted root certificate authority store would solve the problem for Safari. But Apple has been notably quiet. I did a search through Apple

Support for DigiNotar, didn't find anything. I haven't seen any updates. There are, if people are worried, you can put in "Mac OS X DigiNotar," and there are a number of blogs out on the 'Net where people walk you through, walk the user through manually disabling the DigiNotar certificate under Mac OS X. So it is possible for an end-user to do it.

My feeling is Apple must just be on the cusp of releasing a fix to this, and that all Macs will all get fixed very quickly. It's strange that they haven't because they've had a few days now, and everyone's running around like their hair's on fire because this is potentially a problem. We just don't know where the certificate went. Like we don't know how this happened and who's using it and so forth.

So Chromium's source, a reading of Chromium's source, and a tip of the hat to Simon Zerafa, who is frequently sending me goodies in my Twitter stream, and he sent me a bunch of links relative to this that made my search a lot easier, the Chromium source for Google's Chromium browser shows that those three trusted, previously trusted root certificates - the DigiNotar Root, the DigiNotar Cyber, and the DigiNotar Services 1024 - all of those were certificate authorities. They are now - their public keys are blacklisted in Chromium. And what's also really interesting is that there was a sudden jump, and I mean sudden and huge jump, in specific certificates which are blacklisted in Chromium by their serial number. Serial numbers are all unique. There used to be a blacklist of 10 certificates. And they're ones we've talked about before that were signed under a previously found to be fraudulent certificate authority. They're all blacklisted. So there used to be 10. Now there's 257.

Leo: Oh, man. How did that happen?

Steve: And I was looking at a side-by-side, before-and-after source compare. And there's, like, there's these 10, and the same 10, and then this explosion, another 247 in addition. So now there's a total of 257 specifically blacklisted certificates that Chromium's source lists as no longer valid. And this is why I'm suggesting that there's a little more - this is one of the reasons I'm suggesting there's a little more to this story than we've heard yet.

And interestingly, I noted when I was looking at the source that those original 10, nine of those original 10 expire on March 14, 2014. Which reminds us why, annoying as it is, that certificates expire, that is, annoying for webmasters like myself and like you, Leo, who are constantly having to renew the certificates that our servers have and use. The benefit of that, on the other side, is that blacklists don't have to remain in place forever. They only need to remain in place while the timestamp on the certificate would otherwise show that it's valid.

So, for example, in this case, for those older nine certificates that Chromium is currently, every copy of Chromium is carrying around with it, making sure that no one trusts those by mistake, well, those also have an expiration of March 14, 2014. So on March 15, 2014 - and actually they'll probably wait a while because we want to make sure that clocks are correct. But what that does is that allows those to then be removed from the blacklist because the timestamp will show the certificate invalid, and it's no longer necessary to, like, force it in the code on a per-certificate basis. So that's sort of the upside of that.

Now, Mozilla has taken a different approach. First of all, Firefox 6 just updated to 6.0.1 for this change. So Microsoft yanked that out of their trusted root store. Mozilla, that carries their own trusted store in the browser, they updated from 6 to 6.0.1. In the code of this change, the comment in the code says "Do not trust any DigiNotar-issued certificates."

Leo: Any.

Steve: But they're doing, well, they're doing something different. Then down deeper in the code it says "Examine the time window during which the fraudulent certs were believed to have been issued; and, if DigiNotar Root CA is within that window, the user cannot override. Otherwise, if DigiNotar, warn the user, but allow an override." So this tells us a few interesting things. This is that someone - there's been some dialogue somewhere between DigiNotar and Mozilla, and that a window has been established during which this problem occurred. And that this says also that there's reason to believe that more than just *.google.com got loose. Otherwise, all we'd have to do is blacklist that one serial number. Instead, Chromium has blacklisted 247 serial numbers.

Leo: Interesting.

Steve: And Mozilla's approach is also a little softer because they're, I mean, look at the problem that all websites that have certificates signed by DigiNotar are no longer trusted. There's even like a personal identity service, I can't quite remember the name of it now, that the Dutch have, and DigiNotar was a signatory of personally, like personal certificates that citizens were able to get. So those are all no longer trusted. I mean, this is a huge disaster for them.

But Mozilla is softening it a little bit. If it encounters a certificate outside of this undisclosed, well, I mean, it's disclosed in the source code, but we don't know what the real-world implications are. But outside of this time window, then you'll be warned, but you'll be allowed to override, which is sort of nice. It means that Firefox will conditionally trust certificates which are probably almost certainly okay, rather than just lowering the boom the way both Microsoft and Google have chosen to do.

So it's interesting because, from looking at the code, we can't really tell what the story is. But we know that it's more than just *.google.com got loose. There's more story here. And it may be that there's just an embargo on the details until all of these browsers get pushed out and updated; and that once we know that we're going to be blocking certificates, there may be more information made available about exactly what it was that happened. But it does look pretty bad.

Leo: So your sense is DigiNotar is kind of a reliable cert authority. It's not something weird out of nowhere.

Steve: Yeah. They would never have made it into one of the gang of trusted roots if they weren't good. Their site looks very nice. A little hard to read for me.

Leo: Although that's no way to measure reliability.

Steve: I know. But it just has a - they feel...

Leo: They look professional.

Steve: Yeah. And, oh, they're also heavily used by the government there.

Leo: The Dutch government. Oh, that's interesting.

Steve: So there's a huge impact to the Dutch government that has many of their certificates signed by DigiNotar.

Leo: That's interesting.

Steve: And so these are not - this is not a fly-by-night outfit. This is...

Leo: So they got hacked? Or tricked?

Steve: We just don't know.

Leo: We don't know.

Steve: I mean, a trick - I would have to conclude, with the very scant evidence we have, and everyone needs to recognize it's very scant, that this is more of a hack because a trick would be one certificate. A trick would be *.google.com. Whoops, someone somehow managed to trick them into issuing that. But for some reason Chromium has laid out a swath of certs that are being blacklisted. And Mozilla is saying we've got a time window. So again, none of that fits the "We tricked you into issuing a *.google.com certificate."

It's probably going to turn out that something bad happened, and potentially a bunch of certificates were issued. And of course DigiNotar knows, they would have provided this list to Chromium, for example. They know which certificates would have been issued during that time, unless the hack was more sophisticated than we would expect. So I think we'll be coming back to this and updating our listeners in a week or two with more information, probably that a bunch of certs got issued.

Leo: Somebody in the chatroom says they're a division of VASCO, which is a very large...

Steve: They are a VASCO, yes, they are a VASCO company.

Leo: They're an international security company.

Steve: And they're one of the big makers of the tokens. In fact, VASCO is the manufacturer that is often relabeled on all these little footballs that we talk about people press and generate numbers.

Leo: So this, you know, is not some - DigiNotar is not some out-of-nowhere cert authority. This is a biggie.

Steve: Right.

Leo: Wow.

Steve: And they have a nice-looking website, Leo.

Leo: Yes, they do.

Steve: Which is what I go on.

Leo: That's really the only thing you need to do.

[Talking simultaneously]

Steve: Good color scheme. So I was - I skipped over this the first time until so many people tweeted it. And I thought, okay, well, maybe it's of more import than I was originally thinking. And that is the news that Pakistan has sent notices to all of its ISPs, the Pakistani government to all the ISPs, requiring them, the ISPs, to report any use of encrypted VPN traffic within Pakistan. Which...

Leo: Ugh.

Steve: So Pakistan is now formally banning VPN encryption technology. The quote said, "All such mechanisms including EVPNs" - which they called "Encrypted Virtual Private Networks." And of course you don't - you can have a virtual private network without encryption. Most people don't bother, but it's possible. So nonencrypted VPNs are fine. Encrypted VPNs, which is, like, all of them going on, "which conceal communication to the extent that prohibits monitoring." And that has now been banned by the Pakistani government.

All Internet traffic in the country travels through the Pakistan Internet Exchange, which can be and is known to be intercepted by military and civil intelligence agencies. And of course Pakistan said that this is in order to allow them to monitor terrorist activities. And somebody was quoted as saying that the claim that the move is about stemming terrorism "...is like banning cars because suicide bombers use them." So anyway, it's not a good move. But enough people thought that it was important, I mean, I guess I'm not that surprised. It's a rough area to have freedom of speech in anyway. So now there's less freedom than there used to be.

Last week news popped up of a new worm which uses the Remote Desktop Protocol. And what's interesting about it is it doesn't rely on any defects or bugs. It's able to function on fully patched Windows systems supporting the RDP, the Remote Desktop Protocol. The good news is several things. First of all, not that many systems are going to have RDP out on the Internet. For example, ShieldsUP! would tell you instantly if you had port 3389. That's one of the specific ports we check for because I have long known that it's a huge security risk for you to have port 3389 open, which is the RDP port.

The worm that is spreading with RDP just does a standard Internet port scan of port 3389. And if it is able to establish a TCP connection, meaning that there is a listening RDP service at the IP that it has just scanned and found, then it has a dictionary of logins, so it'll attempt to use a username and password login in order to access the server there. And if it's able to, then it essentially uses its access to the user's desktop to transfer a bunch of software across the link and run a copy of itself, which then takes off and begins finding - also scanning the Internet, looking for more copies. It also has bot technology built in, so that it is a bot.

And this was discovered due to a sudden rise in port 3389 Internet traffic. TCP SYNs were being sent out, and we'll be talking about that next week when we discuss what the TCP protocol is. TCP SYN packets began their, like, a much larger flurry than normal because instances of infected systems were out looking for more of them. The reason this is not a huge concern is that, first of all, only servers typically have RDP installed and running. Hopefully servers are behind their own firewall that wouldn't be making the Remote Desktop Protocol public without intending to. The lower end current Windows systems, Windows 7 Home, for example, doesn't even have it. Pro and above have it, but it's disabled by default. And anyone behind a router is protected by the router's inherent NAT layer, which would be ignoring incoming connection attempts, even if RDP was used behind the router on systems.

So I don't think it's a huge warning. I did want to mention it because a number of our listeners had noted it and wanted to know what this meant for them. It's probably not a big deal. But we do have a worm out on the 'Net. And it is significant in that it's not exploiting any deficiencies. It's not exploiting problems. It's just working the way it's supposed to.

Leo: Isn't that nice.

Steve: Just someone said, hey, let's just look to see if anybody's got their Remote Desktop Protocol out and with a dumb password that this thing is able to guess.

Leo: So that's the key. Have a good password, and you're all right.

Steve: Right. I wanted to give our listeners - today's Q&A is almost entirely about Off The Grid. It really captured our listeners' attention and imagination, I think because it's simple. I mean, it had to be simple to be usable. But everyone could easily understand how you could walk around a Latin Square being driven by alphabet characters. And so there was a bunch of really good and interesting questions that will sort of be a nice wrap on last week's disclosure.

I wanted to let people know I'm working on what I call an ultra-high entropy pseudorandom number generator because the one I've got in there now, it's a really good cryptographic pseudorandom number generator, but it's just based on AES Rijndael. As we know, you can generate really high-quality pseudorandom numbers just by using a keyed symmetric cipher driven by a counter, meaning that the counter is - AES is a 128-bit block. So you have 128-bit binary counter, which you simply run through AES or any other really good symmetric cipher, and out comes garbage, gibberish. I mean, yes, you could run it the other way and get the counter value back. But it doesn't matter. Every time you increment the counter, you're going to get a really, a next very high-quality set of bits.

The problem is that, if we used the maximum key size for AES, which is 256, we have 256 bits for key, and we have a 128-bit counter. Which means together the sum of those is, what, 384. So the total entropy, the total amount of randomness that that pseudorandom number generator has is 384 bits. There just aren't any more than that because the way it's generating the numbers are from the 256-bit key and the 128-bit counter. So while that's good for almost every purpose in the world, it's not good for the Off The Grid because we know that there are so many Latin Squares that it's on the order of 2^1418.

Leo: Yow.

Steve: So, and I've always known this, and I knew that I had to replace this PRNG, the pseudorandom number generator because, if you have AES generating your random numbers to produce this grid, and the generator only has 384 possible bits of entropy, you can't produce all the possible grids. Now, yes, you can produce more than we will ever possibly use in probably multiple universe lifetimes. But I wanted to be able to get to them all. And you can't get to them all with a PRNG that only has 384 bits of entropy. So I am developing one which has 1536 bits of entropy, which is to say...

Leo: Wow. This might be of use in many things.

Steve: Oh, it's very cool, yes.

Leo: Is this in JavaScript? What are you writing it in?

Steve: JavaScript.

Leo: Yeah. I hope you release this to the world because I think a good random number generator is worth its weight in gold.

Steve: Well, again, only fools develop random number generators by themselves. So I'm no fool. This is based on some really good technology which has been developed. And I've got links...

Leo: But you're not using Rnd in JavaScript, obviously.

Steve: No. But absolutely, I will have the source code heavily commented and available. I'll be turning it over to the denizens of GRC's Thinktank newsgroup probably later today or maybe tomorrow at the latest because the first thing I want it done is pounded on. I want these guys to suck out a huge block of random numbers and apply it against, for example, the diehard standard suite of random number generators. And, by the way, the core technology of this was designed by the guy who designed the diehard random number generating tests. So it's got a lot of good technology behind it.

But yeah, Leo, it is ultra-high entropy. The cycle time, the time between it cycling back - now in the case of any counter-driven PRNG, like I was talking about, a Rijndael cipher, well, we know what the count, what the cycle time is because we're feeding 128-bit count in, so it's 2^128. This thing is like 2^32 times something or other. I remember what the equation was. I haven't worked it out yet because it's like it'll just, I mean, we really don't need a long cycle time. But I wanted to be able to say that potentially just a ridiculously vast number of Latin Squares could be produced because we have a pseudorandom number generator that has that many bits of entropy. And you need that many that are going to be set in a random state in order to access all of those potential Latin Squares. So it's cool.

Leo: Great, yeah.

Steve: And a bunch of people noted that Becky Worley did a very nice piece on her "Upgrade Your Life" series on Yahoo! News on Password Haystacks.

Leo: Oh, neat.

Steve: Yeah, it's a very nice piece.

Leo: Wow, that's great. See, there's - you see, Becky has an advantage. She's actually somewhat technical. I don't think any reporter in any other sphere would understand at all what the idea of padding the password is and all of that. But she gets it.

Steve: Yeah, in fact, yes, she does. And she went so far as to change some of my examples just, I don't know, for whatever reason, she didn't want to plagiarize. But she changed them and kept them all correct.

Leo: Oh, that's excellent.

Steve: So she did understand all of what was going on.

Leo: Doesn't surprise me.

Steve: And just a little follow-up bit of errata, Stan Robins in Mendota Heights, Minnesota commented about Firefox 5 and 6 and KatMouse. Remember that I had just, minutes before the podcast two weeks ago, I realized the reason my scroll wheel wasn't working on the mouse was that under Firefox 5 that I think I was on at the time, now I'm on 6, I discovered it was scrolling a different tab.

Well, and so he wrote, "Steve, you probably already know this by now," and I did discover this independently. He said, "But if a PDF is open as a tab, using a PDF reader plug-in, the scrolling message via KatMouse goes to that tab. Close all open PDF tabs, and scrolling will work normally on the tab that then has the focus. This is a bug that might never get fixed because the Firefox bug reporting system is pretty lame," he says. Okay, well, I didn't say that, but Stan does. But anyway, the good news is we know that Firefox will be soon rendering PDFs themselves. And I will happily celebrate that day.

And finally, as we commented last week, Leo, when I gave our listeners in real time the link to that cool little $29.95 embedded Cortex M3 development board...

Leo: Sold it right out instantly.

Steve: Instantly sold out. So I contacted the site, the folks there, and said, hey, I'm the guy who's responsible for you instantly selling out of all those. And actually I know that's true because I've been aware of it for months, and they're not selling any. I bought four initially, and then no more sold. And then people were tweeting me after I first mentioned it. And almost one for one, tweet for tweet, I would notice that their stock would drop after somebody would receive the link that I sent. Of course I remembered to tell everybody last week. Instantly they're out. The good news is, from these guys, they told me they're getting 50 units back in stock on September 3rd.

Leo: Oh, great.

Steve: So they will be back in stock. If you can't wait, Digikey and Mouser both also carry it and have them in stock. But I really like these guys at LPC Tools. So they will be back in stock there.

Leo: You know, I forgot to mention that at our grand opening party on the 21st, that Stina Ehrensvrd of YubiKey came to our party. It was so nice to see her. You couldn't be there, but she was there. And we talked about Yubico and all the great things they're doing. They're really - what a nice company. What a nice person she is.

Steve: Yeah, well, and she's moved. She's on the peninsula now.

Leo: She's local.

Steve: Yeah. She's in Northern California.

Leo: And what I didn't know about this whole thing is that there is a kind of subversive point to this. She's a do-gooder. She's more than a technologist.

Steve: Oh, I'm glad she did spend some time with you.

Leo: She's trying to change the world. And I think that, you know, for the better. And I think that's just really neat. So a good person, good company, great technology. And, yeah, I was glad to get some time to talk to Stina.

Steve: Yeah. So just real quickly, this was a little quick blurb about SpinRite that was posted in the newsgroups, in the news.feedback newsgroup. And so that's why Ed says, "Steve, I'm not sure this is the correct place to post this. Apologies if not." And he said, "Today I had my first opportunity to put SpinRite through its paces," he says, "(I've owned a license for a couple of years now) when my girlfriend's laptop went belly up. This was particularly unfortunate timing as she is just completing a course. The exam is on Tuesday, and she was facing the possibility of losing all her course notes plus access to the software she needed to revise and prepare for the exam. Needless to say, SpinRite worked beautifully, and everything is back as it should be. So a massive thanks from both of us." Signed, Ed Metcalfe.

Leo: Very nice.

Steve: And thank you, Ed, for yet again another SpinRite success story.

Leo: All right, Steve. I am ready if you are with questions for the master.

Steve: Yeah.

Leo: Yeah.

Steve: You betcha.

Leo: Questions you've pulled together, I might add. So we won't be stumping Steve on any of these. Well, maybe not, anyway. Starting with Christoph Angerer in Zurich, Switzerland, who writes - he's asking about adding some salt to the grid, for more than just seasoning. Steve and Leo, I just listened to your latest Episode 315. Love the idea of walking through a Latin Square - by the way, you've got to listen to 315, our last episode, to find out more about this - according to the domain name in constructing the passwords on the fly, depending on the path you're taking. My concern is that you effectively change the password security factor from "something you know" to the single-factor "something you have," that piece of paper.

He's right. The problem is, something you have is much easier to steal or copy from you than something you know. Well, so he's not exactly right. If an attacker such as a work colleague, spouse, or friend simply copies your grid, then they can easily reproduce all of your passwords without you ever knowing. The problem, of course, is that your algorithm of how to use the grid is well known. Therefore I suggest you should add some sort of salt to your algorithm. This could be a password that you prepend or append to the domain name in the first phase, kind of like Password Haystacks there. Or it could be some secret change in the algorithm when constructing a password such as, instead of overshooting two characters, you personally, in your own way, always go up four, left three, down one, and then take the characters from there.

Salting adds the "something you know" factor back to your scheme. It will of course not be as secure as a computer-backed hash. For example, if the attacker gets hold of your grid and a handful of generated passwords in cleartext, she could probably reconstruct your salt. However, I think salting still makes the generated passwords much more secure for social engineering attacks. Love your show. Christoph. Well, he got something out of this that I didn't get. Do people know your algorithm automatically? I mean, isn't that...

Steve: Well, they know THE algorithm. That is...

Leo: Yeah, that two over thing.

Steve: Well, yeah. We assume, and all good crypto does assume, that the algorithm is not secret. So, for example, that's the way we've got smart security people checking AES, looking for weaknesses and checking hashes and things. So the concept is that the algorithm is public, and the key that you are using for encryption is private. Well, in this case the key is the configuration of the specific Latin Square which the user has generated and is using as theirs. And as we were just saying, there are so many of them that you just can't brute-force that.

But he's certainly right, and exactly as you reacted, Leo, we've gone from something you know to something you have. So I'll restate again that my goal was to offer something better than what people were using now. And my feeling is that, because this offers a per-domain password, which even those of us who say that's what we're doing, we're probably fudging a little bit on that, I mean, it's just impossible to have a per-domain password for all the different places we go. You know, just to post some random nonsense to a blog somewhere, there may be like a standby, easier password for things you don't need to protect.

So again, the goal here was to offer something sort of simple and fun, which people would actually use, because of course security technology that is not used doesn't provide any security at all. So, and many people, I should say many of our listeners, observed that, gee, this meant that, if someone stole your grid, they had all your passwords. And it's like, yes. So that's a...

Leo: Don't lose your grid.

Steve: Don't lose your grid. Keep it in your wallet and so forth. Now, Christoph is right, though, that I would encourage people to do something custom, do something of their own. They could tack on, prepend or append, some Password Haystack-style stuff, which does not come from the grid. So nobody who had the grid would know. The weakness there is that if someone saw one of your passwords with that tacked on, then they might guess what was going on, that is, like what your salting was. So it's a little better maybe to stick it in the middle or to maybe feel comfortable with evolving the algorithm a little bit. I worked for - I went for something simple and usable. If you'd like a little more security, you could do something different.

I mean, basically one of the things I like about this whole Off The Grid, this whole technology, is that it's just a template. It's a very secure Latin Square that you can use in all kinds of ways. So you could definitely, for example, instead of overshooting and taking the two characters after, you could take the one before and the one after. Or whatever. So, yes, by all means, listeners should feel free to innovate on top of this underlying technology. And you would get some more security against your grid falling into someone's hands if that happened.

Leo: Steve Gowin in Northridge, California wonders about compromised passwords: Thanks, Steve, for all the hard work. I intend to implement this for all of my website logins. But one question, though. What happens if one of my passwords is compromised? Would I need then to create a new password for that site? Or actually he would obviously need to create a new password. The most obvious solution: First, I could keep a list of all the sites that have been compromised and use a different starting point to generate a new password. Because he needs a new password, and he's still got the same domain.

Steve: Right.

Leo: The second option would be to create a different grid and use that. Both have drawbacks. There'd be a need to keep track of what sites he's used the original algorithm and original grid, what sites use the new starting point or new grid. What do you have for me if I have to change a password on a site? What do you recommend?

Steve: Well, okay, a couple things. It has been observed that just starting in a different place in the grid will give you a completely different password. So, for example, in the normal mode we would have people starting on the top line. And I should say that's another way of creating an effective salt is you could start somewhere else always than on the top line, or you could switch to looking up the first character in a row of your choice, rather than in a - I'm sorry, in a column of your choice, rather than along a row. So since it's a 26x26 grid, we've got 52 possible rows and columns where you could look up your first character. But this is handy for people who need an alternative password; or, for example, where policy requires that you change your password occasionally.

Any of the systems that always hash the same domain name into the same hash have a problem that they're unable to do anything else. So this Off The Grid approach does give you the flexibility. So it's one of the things I like about it being on paper, for example, is you could make some notes on the back that this domain, my normal starting place was compromised, so I'm using my backup starting place for the following domains. And it's one of the reasons that the technology I'm in the process of finishing that allows people to reprint their grids anytime is something I think is important because you can imagine over some length of time the back of your grid might get messy with erasures and cross-outs and so forth. And so being able to print a new one and then copy over only the delta information, the little changes that you've had to make over time allows you to sort of keep a nice grid and keep it neat.

I don't think throwing out your entire grid makes sense. Of course that would obsolete all of the passwords that were based on the first grid. And having two seems a little bit of an annoyance, too. It's maybe, you could argue, a little more secure to have a second one. But then you've got to, as Steve mentions, you have to keep track of which one you're using where. So I just think starting, altering your own algorithm some way and then making a note that that's what you've done. And of course you could also - you could sort of just have a standard backup. And if you generate the password for a domain you haven't been at for a while, and it doesn't work, the act of it failing might jog your memory. It's like, oh, that's right, this one uses...

Leo: If that doesn't work, how about three over or something, yeah.

Steve: Yeah, it goes to Plan B. And so then it's like, ah, then it works.

Leo: I do that all the time.

Steve: Yes. I do, too, as a matter of fact, Leo.

Leo: Let's see. Moving along to Question 3 from Rik Schreurs. I'm not doing well on the names today. Rik Schreurs.

Steve: Through no fault of your own, Leo. They're tough names.

Leo: Sorry, Rik. He wonders about an option to decouple the output from the Latin Square. Steve, I thoroughly enjoyed watching your OTG explanation on Security Now!. I've made multiple attempts in the past to create something similar to this, but my attempts always turned out to be too complex or too trivial, either side of the coin there. I've heard of Latin Squares before, and now I feel like an idiot for not coming up with this idea myself. I would add one more thing because, as you said, you can't be too paranoid. Instead of taking the output letters from the grid, you could also add other independent symbols between the navigation cells and use those as output.

For example, between each two square navigation cells, there's a rectangular half cell, say, containing two symbols, let's say a red and a green one. When overshooting the cell with the matching letter, you take one red symbol from the first half cell you jump over and the green symbol from the second. This would totally decouple the password output from the input as the output symbols are independently generated without the Latin Square constraints, and also make it easier to mix in some digits and symbols instead of having to scan all the way to the edge of the grid for those, as you suggest.

Of course you'd have to print the grid a bit bigger in order for room to exist for those extra symbols, and this addition may be too complex for some users. But it would be nice to have something like this as an option in the final program, hint, hint. I'm definitely going to write some code on my own to experiment with this and see if enhancements like the example above are feasible.

Steve: Okay. So it's a great idea. And that was my original idea, in fact. For those who are looking at the video, Leo, if you go to GRC.com/otg/26x26-ppc.png...

Leo: All right. Let's pop that thing up. Holy cow. This has got reds and greens in it, as he suggested.

Steve: Well, okay. So the "ppc" stands for Personal Paper Cipher, which was my original working name before I came up with Off The Grid, which I really like so much. So there's a 26x26. And if you just change it to 13x13 you'll see an alternative. This went through many stages of evolution while I was working to come up with, like, the right compromise between ease of use and security. But you can see from that 26x26 grid, essentially the idea was we take the standard 26x26 Latin Square and interleave columns of complete random characters. And so exactly as Rik says...

Leo: Is that what the green columns are on this image? Or the white columns?

Steve: I think the green are the Latin Square. What's all lowercase?

Leo: The green, okay. So it's green, okay. So green is the actual square, the traditional OTG square. And then you've added, in the white rows, you've added randomness.

Steve: Exactly. So the idea would be, you use the Latin Square as we do now for the navigation, but the output characters you don't take from the Latin Square itself. You take those from the intervening. And I think maybe I was going to go, like, one on each side. So take the left-hand character and the right-hand character that fall on either side of the target Latin Square character.

Leo: Now, these are truly random because I see for instance in the middle column there's three uppercase E's. There's no attempt to make it unique like a Latin Square.

Steve: Exactly. And that's strength. And Rik's point...

Leo: That's truly random.

Steve: Yes, is that an attacker would have absolutely no information about your Latin Square because you're giving none of the Latin Square away. You're generating characters just that are physically associated with it.

Okay, so there are a couple problems. First of all, what do you do if you get one of these annoying websites that won't let you use special characters? And they're out there. There's a surprising number where you still cannot - you only can use alphabetic and a digit and not special characters. So then I thought, okay, well, we could remove the special characters. But there are even some that won't let you use digits.

So I thought, well, they could just, you know, if you run across one of those, you could just skip it, blah blah blah. But the biggest problem is, as you said when you looked at it, it's like, whoa. I mean, it's...

Leo: It's big.

Steve: Yeah. We use the fact that letters are taller than they are wide in order to keep the thing from being, like, really wide. So it's scrunched down a little bit. But it's still, it's twice as much information that we've crammed into this grid. And but here was the key, was with the group in the Thinktank newsgroup, and there are some really smart crypto-oriented people who hang out there, that we've had - we have great conversations. And we carefully looked at what was the nature of the leakage. This is what I call the "structural leakage" that does come out of the grid every time you use it because any attacker would know that there would be the character of the domain name, which we assume they know. And then the two characters that you output, the so-called "overshoot characters," would be right next to it, given that you're using the normal default algorithm that we talked about last week.

So, yes, that does leak a three-character sequence that occurs somewhere in the grid. But I looked at it long and hard before I made the determination that there just isn't useful structure. The idea would be you would apply constraints, those constraints to a grid generator, which would then be responsible for generating candidate grids which obeyed the constraints of all these little triples that you got. But, and this is where the insanity of the number of possible Latin Squares comes to our aid. There are so many of them. Again, it's like 9.333 times 10^436 or something. I mean, that's just a ridiculous number.

And so what I was able to show was that, sure, even if an attacker had a bunch of your domain name and matching passwords, and had all of the triples from those and applied them to constraints on the grid, there's still too much that's left unknown and that you are forced to brute-force. So each one, each little bit of information does leak a little, but there's just - there's so much to be leaked that you really are secure. And an attacker having a whole bunch of your passwords and matching domain names, I mean, that's a weird attack scenario anyway. I mean, we want to understand what the consequences are. But it's unlikely that that happens. I mean, otherwise you've got some other sort of serious problems.

Leo: Yeah. Well, and it's like all the attacks that require people have access to your hardware. I don't worry about those so much because so much can be done if they have access to all your passwords. You've got other problems.

Steve: Anyway, I just want to say Rik's point is right. I mean, and I did want to let people know I went there, looked there, and the two links to those PNG images are in the notes on the site, if anyone's curious. If you actually go through and read all the text, I do address that issue.

Leo: Great. Brent Nesbitt wonders if Off The Grid goes both ways. Is it possible to decipher a message if you have the grid that was used to encipher it? If so, what's the process?

Steve: Okay. This is such a neat question, I am posing it to our listeners.

Leo: A stumper.

Steve: Can you, and under what conditions can you, decrypt from the password back to the domain name? I mean, it's a perfect little puzzle for everyone to think about.

Leo: I like it.

Steve: So I'm not going to answer Brent's question. I'm going to say, well, what does everybody think?

Leo: And how would people respond to you? Go to GRC.com/feedback? Is that what you'd recommend?

Steve: Yeah, or tweet.

Leo: Or tweet. @grc, that's Steve's...

Steve: @SGgrc.

Leo: I'm sorry, @SGgrc, that's Steve's handle. Good. Can you go both ways?

Steve: Yes. Can you, given a password, do you have enough information to figure out what the domain name was? And because you could find those - we know that the password is character pairs that occur in either horizontal or vertical relationship to each other. And then you'd see the other character pair. And so one's going to be horizontal; the other's going to be vertical. And it's wonderful to think about, so I'm going to let our listeners think about it.

Leo: Think about that.

Steve: Think about that.

Leo: Think about it. Billy Spelchan, Billy D. Spelchan has been thinking about how the Latin Squares are generated: When I was watching your Latin Square generator working - which is fun. Turn that on. It slows it down, but it's so fun to watch it go [vocalizing]. I noticed that you generate the rows randomly. I'm assuming you're using data from the above rows to determine valid letters, then randomly choosing from those. When you run into a non-solvable cell, you then back-step - and you'll see that sometimes it goes [vocalizing] - and try with different letters.

Steve: And it makes just that sound, Leo.

Leo: [Vocalizing]

Steve: You can almost hear it.

Leo: And this is just because you're doing it by brute force; right? You're not being smart about it. You just go and, boom, let's make it work. And it's fast enough that it's fine. In fact, you probably do use a more powerful algorithm, I would guess, to just do it.

Steve: Oh.

Leo: Yeah. But I like it when people think about this stuff. Wouldn't it be easier to create a basic - each row shifted over one - Latin Square and then do a few dozen iterations of randomly swapping each row with another random row and each column with another random column? So you've solved it in one direction, and then you swap it around. This would be easy to recreate if you used a cipher and a sequence of numbers for getting the random numbers for the rows and columns to be swapped. I'd love it if you explained your choice of and design of Latin Square generation algorithms. And actually you didn't really talk about how you did it in the fast technique.

Steve: No, no, I didn't. Okay. So the algorithm, and I will have - I'm going to release the source code to all of this as soon as I get it finished. There's been a bunch of people asking if they could implement it in little utilities and apps and smart phones, which just delights me. So absolutely, I'm going to encourage that, and I will help that effort by letting people all have my well-commented JavaScript source as soon as I get it done. I didn't want to let people run off half-cocked before I'm finished because I'm still making, like in the case of this ultra-high-entropy pseudorandom number generator, which is the key of being able to recreate these things, I'm still making some changes. So I've got the code stripped of comments and obfuscated just while I'm getting it finished.

I'm really pleased with the algorithm. So think about going across, filling in the first row of the Latin Square. For the very first character, it could be any one of the 26 characters of the alphabet. The second one could be any one of the 26 except the one we already have because we can't have two in the same line. The third one could be any of the remaining 23, and the fourth one any of the remaining 22, and so on. So as we move across, what the algorithm does - and the implementation came out really nicely because I use bitmaps to represent the characters which have been used so far. So as we're moving horizontally from left to right, I'm keeping track of the characters that have been used behind us on that line, and then selecting at random - and so there's the key, selecting at random - from the set of remaining possible characters we haven't yet used on that line.

So the first line, zip, I mean, nothing ever stops us. We just go right across and in some random sequence we lay out all the 26 characters of the alphabet. But now we're on the second line. So the first cell of the second line has to consider what's above because we can't have any duplicates down that column, that first column. So I also have bitmaps which I maintain for all of the columns. And this is where bitmaps are so cool, because I'm able to AND the current row's bitmap with the current columns bitmap in order to get, instantly get that subset which doesn't appear either in the column so far, or in the row so far.

So now we move along again doing what we were before, but also considering making sure that we don't have a collision of anything above us. And even there it is possible that we could make a wrong - that randomly we would have chosen something where we would use a character on the line incorrectly, where we really need that character to be used later and a different character that would have been left over later used earlier.

So what happens is it gets to the point, it can get to the point where it's on a cell, and it does the ANDing of the two bitmaps, and it's zero. That is, there are no available characters which, due to the choices we've made before, which don't occur, that we haven't already used, either in the row behind us or the column above us. In which case we backtrack. So this algorithm is a pseudorandomly driven, back-tracking tree search because we're searching through a tree of possible Latin Squares. And when we backtrack, we remember what didn't work. So that when we back up a cell, a memory is kept of the choice we made that time, and we know that since we're coming back, there's no solution downstream of that choice. So that choice is removed, and the pseudorandom number generator is asked if we've got any more available characters. And that's the way this proceeds.

So it just - it works its way down, line by line. And of course, as we get down further, it becomes increasingly difficult, which is why you'll see the generator just cruises along without much trouble at all toward the beginning of the square. But around about halfway down, it starts choking a little bit because it's got so many characters above it in each row, and so it knows it can't choose any of those. And it's also having, as you'd expect, more trouble toward the end of the lines because it's got more characters that it's already chosen behind it that it can't choose ahead of it. And the other really interesting thing is the last line is a freebie. So if you think about it, every character in the last line is predetermined because it's all of the characters that didn't occur in the column above that cell. So that one we get for free.

One of the things that I noticed, though - and, I mean, I watched this thing obsessively for quite a while, and it was tricky to get it to run as fast as it does, actually. I had it running at 13x13, and for a while I was thinking I was going to use a 13x13 Latin Square and just double-up characters in each cell in order to keep the grid size down. And then, expanding it to a larger one, it became tough to get it to run fast enough. And I did expand it up because I just wanted all of that entropy that a full 26x26 Latin Square has.

What I noticed was that sometimes it could make a mistake early in the line. It could choose a character, like in the fifth cell from the left as it's moving from left to right, which is wrong. Yet it could go way downstream, down that line, and just never be able to solve that line. It would just - and essentially it would be backtracking, choosing a different character, and then trying to move forward again, backtrack again, choose a different one, then go back even further because that, I mean, the point was I realized an early mistake could take us a long time to recover from.

So there's two modes in my generator which you can choose on the website: Watch It Work or Get It Done. The Watch It Work uses the algorithm I was just talking about for generating the square. But it can get stuck and take a long time. It's kind of cool when it does because you can see it, like, changing its mind and doing what it can. The just Get It Done is a different strategy that I developed from watching it get stuck otherwise. And that is, the first instant that this thing has to backtrack, it just scraps all the work on that line.

Leo: Oh, it starts over.

Steve: It starts the line, yes. And for whatever reason, I mean, and actually when you think about it, for it to be able to fill out a Latin Square as easily as it does, it never, for example, goes back up into earlier lines. It has the ability to do that. It can backtrack all the way back out to the beginning, if it had to. But the fact that it never has to go back up to a prior line means that there's always a solution downstream. Which gives you some sense for how many Latin Squares there must be, if it's always able to work itself out.

So anyway, that's the algorithm. Now, Billy suggested something very clever, which was think about a - here's a trivial Latin Square. Let's just think of it as maybe a 10x10, and we'll use digits, so 0 through 9. We fill the first row with 0 through 9. Then we fill the second row, shift it over. So we start with 9, then 0 through 8. And the third row we shift again, so it's 8, 9, and 0 through 7. Well, if you keep doing that, think about it. You've just made a Latin Square, a trivial one. Because obviously all the lines are going to be okay because they're all only 0 through 9. There's no attempt to repeat there. And you've automatically fixed the rows because that skew means that there'll be a different digit automatically in every column space.

So as Billy says, that's a simple Latin Square. Then we already know that swapping rows and swapping columns never breaks the Latinness of a Latin Square. If you think about it, if you swap rows, well, the rows are all going to still be fine, and the columns will be fine because you're not creating any duplicates that didn't exist before. So that kind of random transposition of rows or columns always is safe to do to a Latin square.

Well, that is the genesis of the Latin Squares Workbench. And in fact, in the podcast last week I said "latinsquare.htm," that is, GRC.com/latinsquare, and I think it's plural. So some people tweeted they couldn't find latinsquare.htm. It should be latinsquares.htm. And that's this little Workbench that I wrote, also in JavaScript, to familiarize myself with manipulating Latin Squares.

My original concept was, boy, it's exactly as Billy suggested, it would be so easy to generate Latin Squares that way. And in fact one of the - rather than randomly exchanging them, what I would have done in a more robust crypto standpoint would have been, once I had that simple rotated or shifted rows Latin Square, then I would have taken another, like a blank Latin Square, and randomly have chosen one of the rows and stuck it in the first row of the new one, randomly chosen one of the remaining rows in the original one and stuck it in the second row of the new one, and so forth.

What that means is, what that says is that I could choose any one of 26 rows in the first one to be the first row of the second, any of the remaining 25 from the first one to be the second row of the second, and so forth. Meaning that there are 26 factorial ways of rearranging the rows on a 26x26 grid. Similarly, there are 26 factorial ways of rearranging the columns. So what that says is that there's 26 factorial-squared arrangements. Which is a huge number.

But it turns out you cannot get to all Latin Squares that way. It's huge, but it's not huge enough for us. It's not all possible Latin Squares. It turns out that there are - I think they're called "paratopy sets." The Wikipedia page on Latin Squares discusses this, where these are like closed sets of Latin Squares which are large, but they're disjoint from other Latin Squares. So you cannot get between any two Latin Squares just by swapping rows and columns. And that I proved.

I wanted to see that for myself and develop a feel for it, so I wrote that Latin Squares Workbench to actually play with swapping rows and columns. And that's what that is. That page on GRC, latinsquares.htm, allows you to, like, with your mouse, click on rows and columns or even click on symbols and swap them and experiment with different configurations. So anyway, I wanted - you know me. I wanted, if I'm going to do it, I'm going to do it once and do it right. And so having an insanely high entropy pseudorandom number generator that potentially lets us get to at least as many Latin Squares as we know exist, is the solution that we'll have shortly.

Leo: Very cool. Really neat. Let's see here. Moving along to Question 6, Drew Monrad wonders about the Off The Grid source code: I am just starting to take a look at iOS programming. I thought I'd like to try something related to your OTG, make an interesting starting challenge. I think that would be neat, to have an iPhone app out of this. I don't know how much I'll be able to get done. Are you able to share your JavaScript coding? This would let me concentrate on the Objective-C side of the project. This is more of a personal challenge than an attempt to release a new app. But if it turns out to be worth sharing, I'd consult with you before publishing anything with Apple. Thank you.

Steve: Yeah, so I say to Drew and all of our listeners, yes, absolutely. I'm going to let everyone have the source code for this as soon as I get it done. And, frankly, I love the idea of turning this into some smartphone and, like, little standalone utilities because what that lets us do then is we kind of get the best of both worlds. We could keep our grid in a drawer or in our wallet as the master paper backup reference. So we could generate a password with zero technology if we wanted to.

Yet if we had a little app, it would make it much easier to just type in "Amazon" and, boop, there's our 12-character matching password. So we have a piece of technology that makes it easy, but with the caveat that, if it's online, if it's in your phone, I mean, it's potentially vulnerable to being compromised. But with the understanding of that tradeoff, then, yeah, I think it's a cool little project. And I will happily provide all the source to help people make that happen.

Leo: Very kind of you. And the truth is, even though it's obfuscated JavaScript, it's fairly easy to use tools to get that source code. But I'd rather see Steve's commented source code than anything else, obviously.

Steve: Yeah. I had somebody sent me a note who said that he looked at my JavaScript. I think it must have not been this project because I think it must have been something else I had done because I think only one file is currently non-obfuscated. And he said, "Steve," he said, "I hardly even needed to read the code, your comments were so good." And actually I learned the lesson a long time ago. I write this stuff for myself mostly because I know I'll put it down and won't come back to it for a decade. And then I'll be thinking, what the heck was I thinking?

Leo: Well, if you're an assembly language programmer, commenting is essential.

Steve: Yeah.

Leo: I mean, you really learn that, I think. But, I mean, it is for every programmer. But assembly language, you...

Steve: Well, actually these algorithms will, that Latin Square finder, people who enjoy code will love looking at it because the way I implemented it with these bitmaps, it's like there's almost nothing there. It's like, wait a minute. Where's the code? I mean, that little blob generates Latin Squares? It's like, yeah. It ended up being really good.

Leo: That's neat. That's excellent.

Steve: I'm proud of it.

Leo: Yay. You're a good man, Charlie Brown. Question 7 is from Justin Lowmaster at TheSpaceTurtle.com. It's in Portland, Oregon. And he has a special case for what we were talking about, just why would you need WAN administration on a router? Just turn it off. Steve, regarding the Universal Plug and Play UPnP exposure and WAN-side administration: When I was "dating" my - he puts that in quotes, I don't know why. When I was "dating" my now wife, she lived all the way across the country and three hours later in time zones.

Steve: Oh, that's why.

Leo: That's why.

Steve: Because that was sort of virtual dating.

Leo: Not much of a date.

Steve: Yeah.

Leo: As the night wore on, her Internet performance would get flaky and drop, needing a reboot of the router. Well, obviously they were heating up the router a little bit. Unfortunately, the router was located in an area that would cause her parents to get annoyed if she left her room to go reset it. My mind, I'm sorry, it's just bad. Okay.

Steve: Yes, Leo.

Leo: I got her to turn on WAN administration, but only allowed for my IP - and any malicious people who knew a workaround, but I was young and in love. This allowed me to remotely log into her router and reset it without waking Mom and Dad, and soon she'd be back online, and we'd be getting back to chatting. It might not have been the most secure thing to do, but I certainly think it was worth the risk. Now if I could only find a way to remotely log into our two kids and adjust the auto sleep timers on them. Isn't that cute? That is adorable. He's doing a very strange smiley face that I don't really - I don't understand. Maybe...

Steve: He must have a mustache or something, I don't know.

Leo: Oh, maybe that's it. Maybe the chatroom could explain what this is here. What is that? What is that? It's a brace...

Steve: I guess the colon is his eyes. So he's got hair and eyes.

Leo: He's wearing a Mexican hat. And a beard.

Steve: Oh.

Leo: That's what it is. He's wearing, okay, he's wearing a sombrero. He's going, "Oh," and he has a beard.

Steve: Okay.

Leo: Or else he's got a really big nose, and he's sticking out his tongue. But those are the two choices. Love the show. Own SpinRite. Recommend it to anyone who says "hard drive." I'm still young and in love, but I have safer routing settings now. Well, he's right, that would be a use case, so to speak.

Steve: So I did want to mention that a number of people suggested that the reason Universal Plug and Play was open on routers is for ISP administration. And it's like, oh, goodness. Okay. Again, it's such a horrible security exposure. Now, what he did, which is limiting access by IP, is actually very good. It's still a little nerve-wracking to have something that could listen. But as we will learn next week in our How the Internet Works continuing series, this time on TCP, the Transmission Control Protocol, filtering by IP is very good because nobody coming from any other IP or even spoofing his would be able to access his router, assuming that it was over TCP. And that's important.

Universal Plug and Play is a UDP protocol, and it is possible to spoof that source IP so that the packets would appear to be coming from, for example, Justin. However, if the protocol, that is, the UPnP protocol itself required some transaction, some interaction, then the router would send them back to Justin, that is, to the apparent source IP, not back to the attacker. So again the attacker would be unable to do what they wanted unless a single packet was able to accomplish their nefarious purposes, in which case UDP could have its source IP spoofed. But otherwise it's pretty secure. And I did want to note that, to remind people again, to shut down WAN-side UPnP administration. And if your ISP says, well, we need you to turn it on so we can get in there, it's like, well, then turn it on for that purpose, but then turn it off again. You just don't want to fly with it open.

Leo: Or fly with an open fly.

Steve: With your open fly. I was thinking the same thing, Leo.

Leo: There's a joke there somewhere. I was just too slow to pick up on it. Number 8, Caio Katayama in Hyannis, Massachusetts, corrects you about LPCXpresso's IDE: Steve, just wanted to let you know that the LPCXpresso IDE does not support Mac OS. Windows and Linux only. Aw.

Steve: I thought that that was important, so I had said last week, Windows, Mac, and Linux. And I just assumed that, if they were doing Linux, they would have done Mac first.

Leo: Well, you said it was Eclipse; right?

Steve: Yeah.

Leo: Well, Eclipse works on the Mac.

Steve: Okay. So maybe it's possible to make it happen. They may just not offer it by default.

Leo: Yeah, I mean, I could see they might have an SDK that doesn't work on the Mac. That's probably what it is is that whatever the software development kit is doesn't work on the Mac. But I...

Steve: It's all just C, though. It's all just C library stuff. So...

Leo: It's generic C.

Steve: ...maybe if someone has it running, or is able to run Eclipse on their Mac, they'll be able to do this, too.

Leo: I've run Eclipse on my Mac many times. I still do. It's how I do the Android development. But...

Steve: Ah, but it does use USB. So there will be a USB driver, probably.

Leo: Might be a USB driver. But that's not a big deal. I think that's doable. I think you could figure it out. The other thing is to make sure - sometimes people run up against this. The Mac does not install its developer tools by default, which means you won't have a C compiler on the Mac. So that could be that maybe he needs those, and that's a new install.

Steve: I think it brings GCC along with it, though. I think it...

Leo: Okay. Yeah, Eclipse has GCC, yeah.

Steve: Yeah.

Leo: I don't know. Okay. Well, take your word for it. Something to look into.

Steve: But I did want to alert listeners that maybe not the Mac.

Leo: Right. Scott Maser in Colorado Springs wonders how many? Steve, during your Off The Grid podcast you mentioned that nobody knows how many Latin Squares there are that are greater than 11x11 because it's just impossible to compute. Later on you went to say that there are at least 9.337 times 10^426 possibilities for 26x26. If you can't tell how many there are for a 12x12, how can you tell for 26x26? I'm just curious. How do you do it, Steve? Whatcha looking at?

Steve: I was looking around for the book because I bought a very expensive huge combinatorial math book...

Leo: Oh, my god. You are such a geek.

Steve: ...at the beginning of this because I wanted to understand this stuff, too. And I absolutely can't. So here's what I know. And Wikipedia again has a nice coverage of this, but not the derivation of the math. And it was by following the links that I found this textbook, and I plowed into the textbook, and I got in a few pages, and I just thought, okay. And then a miracle happened, and here was the result. What we know is there is a formula which has been arrived at which I show on the Off The Grid pages at GRC, where mathematicians that have studied this - and, I mean, Latin Squares have really intrigued mathematicians powerfully for...

Leo: I can see why. It's fascinating.

Steve: Yeah, it really is interesting.

Leo: It's a simple concept that's got incredible combinatorial issues and powers and so forth. And look how much we love Sudoku, which is just a special case of this.

Steve: Yes, yeah. And as you said, Leo, that's a perfect way of phrasing it. It's so simple to say, not have anything repeat in any row or column.

Leo: I got it. That's easy.

Steve: Then look what happens. What falls out of that is many, many grids will have duplicates. Some won't. And so the natural question is, well, okay, how many don't? Well, it just - we don't even, in this day and age, have the math for describing it. But the mathematicians have watched the way - they've looked at simpler squares and worked to understand why simpler, smaller Latin Squares, how they function and what can be known about the smaller ones. And they've arrived at two different formulas, one which says we know that there are fewer than this many, and another formula that says we know there are more than that many.

And it's that second formula, we know that there are more than that many. They've been able to say, we've been able to prove that a certain n-by-n Latin Square will have at least this many. And so there's a floor that they've been able to establish. And that's the number I've been using. That's that 9.333 or 337 or something times 10^436, which the log2 of that, that is, the number, the equivalent number of bits, is 1418. So 2^1418. We know there are probably way more than that many, but at least that many. And I thought, okay, that's enough. That'll be all.

Leo: Okay. That'll do. That's all I need.

Steve: Yeah. Because, I mean, consider that most of the crypto we're dealing with is 128 bits. And so this is 1418. And the pseudorandom number generator that I'll have soon is working from an entropy pool of 1536. So, yeah, we're okay. So anyway, we don't know how many. We just know that the mathematician gods tell us it's...

Leo: It's a lot.

Steve: ...at least - it's a lot. It's more, it's so much ridiculously more than we will ever need. But that's where we get security because, even though this is leaking a little bit of structure, it turns out that there's just too - even limiting them in half or, I mean, like even hugely limiting them, down to a few percent, even a few percent is still ridiculously impossible for a bad guy to brute force. And that's the key.

Leo: That's the key. Walter, I'm sorry, Willem Jan - let me do it right. We have such an international audience, I love this. Willem Jan Gerritsen in the Netherlands is chafing under his company-enforced password policy: Steve, in our company we have an enforced password policy, such as every eight weeks we have to change the main password. It can't contain your username. It must contain digits, upper and lowercase, and non-alphabetic characters. And it has to be at least eight characters long. All of which sounds like good password policy. But since we have to change it so often, almost everybody is using an incremental number in his password.

Oy. So they're having the first name, last name, and then 46, 47, 48. What I learned from your Haystacks story is that a long password is what really helps. And eight isn't very long, is it. How does a password refresh policy make passwords stronger? I suspect it just drives people to put it on paper on their desk. In fact, this has been a whole discussion. I think it was Bill Gates who brought this up, who said the problem with strong passwords is people write them down and put them on Post-it notes.

Steve: Well, and I have to say there was an article floated around a couple of weeks ago that someone was claiming that, in a survey that was done, that in corporations the IT people were the most disliked.

Leo: Yeah, I think that's probably true. I hate to say it.

Steve: And this is why, Leo. I have to agree with Willem, I do not see a benefit. If you, for example, have a stronger password policy, like come on, at least eight characters, as you said, that's just not long enough. We need 12. It would be better to have 16. Come up with, you know, like, give everybody a password school as we do here on the podcast. Explain to them what's necessary. Choose a really good password once, and something you don't have to write down, so you don't have to worry about it being stolen.

I mean, I don't get this change it every eight weeks. That doesn't fit the model of exploitation. It's not as if passwords are traveling by camel after they've been stolen, going to the bad guys, and so there's, like, some weird eight-week window, like, oh, we're going to change your password so that the stale password no longer works. Well, passwords are used instantly. Sometimes they're used on the fly, between keystrokes they escape, and they go beamed off somewhere else and then someone's logging in as you immediately.

So this, I don't get this change it every week. And all this does is make IT people despised because users, who are not dumb, they think, why am I - why do I have to do this? What problem is this solving? And when you make people take their shoes off to get on an airplane, all you do is - that doesn't have any clear security benefit. You just upset people. And this nonsense of forcing people to change their passwords continually is exactly like that.

Leo: Yeah.

Steve: So I agree completely. I don't see any benefit to it. Have a good policy and just let people come up with a password that they can stay with. Otherwise you're just asking for trouble, I think.

Leo: Our last question, I think. Yes? Eleven. James Higgins...

Steve: It's an odd number, but then it's prime, Leo, and...

Leo: It's prime, and we love that.

Steve: Yes.

Leo: Speaking of primes. James Higgins in Gadsden, Alabama. He's worried about the UPnP vulnerability we discussed: Steve and Leo, I can't say I'm a long-time listener. I've only been listening for the last few months. But I love the show. Good, because you know there's, like, how many? Plenty more, 315 previous episodes. You have plenty of listening ahead. I was listening to 315, and the part about UPnP really caught my attention. I'm just wanting to confirm if this is firmware-related or something involving a piece of hardware. Reason I ask is that I, like many other listeners, use one of the open source firmwares, DD-WRT being my poison of preference. I like that, too. Can you please confirm where this vulnerability lies in these routers? And thanks for the great show.

Steve: It is absolutely in the firmware, meaning that DD-WRT does not have this problem.

Leo: Good to know.

Steve: So if you took a Linux router, which does - Linux. I'm sorry. If a Linksys, a Cisco Linksys router...

Leo: Well, ironically running Linux, so you're all right there, I think.

Steve: Yeah. But which does have UPnP exposed, and you reflash the firmware with DD-WRT to give it a much beefier, nicer, feature-powerful router, then this problem goes away in the process.

Leo: Right, right.

Steve: Just as James was hoping.

Leo: Yeah. I love it. Tomato, too. Those are both great router firmwares. You have to have the hardware that the firmware works on, just like putting a ROM, a custom ROM on your phone. You have to make sure that they match. But if you've got the router that does it...

Steve: Right.

Leo: As my mom says, the "Linksies." I have the "Linksies" router. Hey, Steve, always fun, always informative. Just love the show. And I'm glad we got so many questions about OTG because I think it's just really a great and fascinating subject and a neat technology.

Steve: Well, I'm just glad that it's out there. Now it exists. It's simple to understand. People get a kick out of it. And we have a paper-based cipher. I just wanted there to be one because I looked around and there just - no one had done one. So, and now we did, and I stumbled on the idea of the Latin Square as a means of maneuvering and creating state, so that you end up having a dependence among everything that's come before. So it solves the criteria of being a useful, paper-based cipher. I'll get the pages finished and updated, and then on to my next project.

And next week, speaking of which, we're going to, as I have said, plow into TCP. I don't think we'll be able to do the whole thing in a single podcast. TCP, there is so much going on there, and it is so cool, it's probably going to be worth giving just that one protocol, which happens to be the most used protocol on the Internet, its due.

Leo: And by the way, we have a stumper. We're asking if you can figure out, is OTG, Off The Grid, is that magic square reversible? Can you decipher as well as encipher using Steve's OTG grids? And you can mail your answer, email your answer, well, just go to the website, GRC.com/feedback, and fill out the form with your suggestions. Chatroom already says they know.

Steve: Well...

Leo: You're so smart, NinjaHacker. If you're so smart, just send a note to Steve and tell us why. NinjaHacker says, "I know."

Steve, thank you so much. We do the show every - now that we're back on schedule, Steve's back home we do it every Wednesday, 11:00 a.m. Pacific, 2:00 p.m. Eastern at live.twit.tv. You can watch, of course, live. Love it if you do. I don't have to say "live.twit.tv" anymore, by the way. With the new web design, just TWiT.tv. There is - I don't know what happens if you go to "live" anymore. But it's just TWiT.tv. Live's right on the front page there. Just click the Watch Live button. But you'll also see there the Security Now! link, and you can go to the Security Now! page and get all the previous, all 315 previous episodes, and this one, too.

Now, Steve maintains an archive of low-bandwidth versions, 16KB versions, as well as full typed out transcripts by a human. So that's another resource, and that's all at GRC.com. While you're there, check out SpinRite, the world's best hard drive maintenance utility. It's Steve's bread and butter, so let's support Steve by all buying a copy of SpinRite. What do you say? GRC.

Steve: And I did want to mention that the Off The Grid section has its own feedback page, which a lot of people have been using. And that helps me because it does then segregate the Off The Grid-related feedback from the general Security Now! feedback. So

Security Now! feedback is GRC.com/feedback. But if you go into the Off The Grid pages, you will see a Send Us Feedback link, and there's a special little form there just for Off The Grid. And it comes into a different email account for me and allows me to see those coming in.

Leo: That's great. That's great. GRC.com, it's all there, my friends. Thank you, Steve. We'll see you next week on Security Now!.