SECURITY NOW!

Transcript of Episode #310

Listener Feedback #122

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-310.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-310-lq.mp3

Leo Laporte: This is Security Now! with Steve Gibson, Episode 310, recorded July 19, 2011: Your questions, Steve's answers, #122.

It's time for Security Now!, the show that covers your, yes, your, yes, your security and privacy online. And what better person to do it than this cat right here, Mr. Steve Gibson of GRC.com, the Gibson Research Corporation, creator of SpinRite, world's finest hard drive maintenance and recovery utility, creator of many free utilities for us all, in the realm of - hey, Steve, how are you?

Steve Gibson: Hi, Mom.

Leo: Hey, hey, how's it going?

Steve: And we also talk about other people's security, or lack of.

Leo: Yeah.

Steve: As it bears on us and our Internet experiences. And this week we've got a Q&A, our hundred and secondy second.

Leo: Secondy second? Sounding like Bilbo Baggins here.

Steve: 120nd.

Leo: My hundred and secondy second. Actually I'm excited because this is our last Security Now!, should all go well, from the TWiT Cottage. Next week we will be doing this show - I think, I hope, well, I think we have to...

Steve: In your new digital studio.

Leo: Yeah, in the new studio down the road in the TWiT Brick House. I think we have to because I think that Sunday - and you're coming up, by the way, thank you so much.

Steve: Yeah, that'll be a blast.

Leo: The grand opening of the Brick House is this Sunday, July 24. We're going to do a special live TWiT with four of the earliest TWiT cast members, all four of whom appeared on the original Screen Savers. And I think when you get in the studio you'll see why I wanted that crew because it really feels and looks like the Screen Savers. Steve Gibson, John C. Dvorak, Kevin Rose, and Patrick Norton will be in the studio with us at the new TWiT Brick House this Sunday. And then the following Wednesday we will do Security Now! as we usually do. What is unknown is whether we'll be in my office, because you know we built a duplicate of this studio. But I don't know if it'll be ready yet. If not, we'll be at the roundtable or somewhere.

Steve: Or any of the new multiple venues you have in this amazing new creation of yours.

Leo: Oh, it is so gorgeous. I just can't wait till you see it. No onscreen webcam or pictures do it justice. Wait'll you see it live on video. You'll flip your lid.

Steve: Have we ever talked about, or recently talked about the URL, so that our non-live listeners could tune in and just see the webcam of the studio as it's being built?

Leo: No, let me give it to you. Because, I mean, it's not super high quality. It's a drop cam.

Steve: But it's better than nothing, which is what a lot of people have right now, probably.

Leo: Right, and it's live, which is kind of fun. I think right now, let me just check, but I think right now there are people measuring for stuff. If you go to bit.ly, it's a shortened URL, bit.ly, bit.ly/twitdrop, you could see what's going on over there. And it looks like there's - oh, oh, the red - a few of the final touches are coming. These are the - these red kind of arches go on the metal arches. These aluminum arches, we've been waiting for those. I think there's a logo that's going in my office. I can see, you know, they've covered - this is the switcher here. They've covered that up to protect it. Last night John and Burke and Colin were all at the studio well past midnight setting up audio. And apparently John's got it all done, which is fantastic. We used that IP-based audio solution from Telos, the Axia, which is spectacular. But it does require a considerable amount of setup, you know, you have to - once you program everything in, though, it just is a button push to change configurations.

Steve: What are those wacky things that you've got the arches terminating into? Looks like sort of some little droid sort of...

Leo: Isn't that funny looking? You know what they really are is Rubbermaid tubs painted copper. But don't tell anyone.

Steve: How fun.

Leo: And they've made some lids for them. The idea was to kind of have some sort of termination point in there.

Steve: Yeah.

Leo: Roger, our set designer, Roger Ambrose just is an artist. He really is an artist.

Steve: And what's covered up there in the foreground?

Leo: That's the switcher. This is where the...

Steve: Oh, my god, that's like some big, like, studio desk kind of thing.

Leo: You'll feel like you're at the bridge of the Enterprise. You're Mr. Sulu sitting here. Everything is there: the audio mixer, the light mixer, the TriCaster mixer for video, all of the Vidyo connections. By the way, we'll be using Vidyo, not Skype, although we'll have that capability, as well. The screenshot station, too. One person will be sitting there. And that rotates. It's covered up, obviously. But that rotates to point to whatever set we're using. So you could be facing...

Steve: How do you get all the cables into it? Do they come in down from above?

Leo: Well, this is the most amazing thing. This building, it's as if this building were designed for what we're doing. You know, a lot of studios have - a lot of electronic installations generally have raised floors. You know the old IBM mainframes, they'd have a raised floor, and all the wiring would be under the floor. Well, this place has essentially a six-foot raised floor. It's our basement.

Steve: And you can drill holes through it?

Leo: We drilled holes all over this floor.

Steve: Oh, my god.

Leo: Every desk, every set, everything has a hole drilled through the floor going down. And so all of the noisy, hot electronics for the mixer, the lighting board, the audio, the video, the Skype even, the Skypesaurus, Skypesaurus, everything are underneath the floor here.

Steve: How cool.

Leo: It makes it so easy. And it's so much...

Steve: And it's concrete flooring?

Leo: No, it's just a wood floor.

Steve: Oh, okay.

Leo: You'll see when you - I'll give you the grand tour, of course. In fact, when we get over there we'll have a live camera, and we'll go all over the place, show everybody what we've done. But having that in the basement is huge. It makes it so pleasant and quiet upstairs in the studio area because all the noisy hot stuff is downstairs. We're building an enclosure for it and an air conditioning system, as well, so it won't overheat down there. Although it's pretty cool. It's a basement.

Steve: Wow, how neat. Well, so bit.ly/twitdrop...

Leo: Yup, that's what you're seeing right there.

Steve: ...will allow our listeners to take a look at what's going on at the studio, those who haven't been watching live.

Leo: Live and in person.

Steve: Nice.

Leo: And it's, you know, we've also done - John has been doing this whole time, for six months now, a time lapse. So there will be a massive, probably a five-hour movie, the making of the TWiT studio.

Steve: With everything like - oh, there's a big red beam moving right now.

Leo: Yeah. So that's going to go on these arches. Those are going to go on the arches to - the arches are very aluminum looking. These are actually lighting trusses. And so to make them look a little more like the Golden Gate Bridge, kind of, this is all International Orange, kind of to make it more structural and less aluminum-y, we're putting these facades on top of them. And it really - just wait till you see it. It's kind of hard to describe how beautiful this is. But the time lapse will really do a great job because you'll see it go from just a boring kind of office with drywall to this.

Steve: Yeah, you're going to want to do, like, a formal introduction to the studios that you can post on, like, statically post.

Leo: Absolutely. We have - well, yes. We have lots of plans for making of, how we did it, all that stuff. Because you know me, I don't like to keep anything a secret. I want everybody to know how you do it in case they want to do it, too. First thing, get a really big bank account.

Steve: And not be in a big hurry.

Leo: And then, yeah, not have a deadline.

Steve: Yeah. In fact, I just - I corresponded with Eileen this morning, saying, okay, is this really happening on Sunday? Because I built things once, and I know that schedules are subject to change.

Leo: It has to.

Steve: Yeah.

Leo: It has to. We printed stickers. With the date on them. It has to.

Steve: Cool.

Leo: I said this a couple of weeks ago. I said, you know, we've printed stickers with the date on them. This is going to be humiliating if we don't actually move on that day. And Lisa said, no, no, we're going to move on that day. And you know what? I think we are. I don't know who this is, here. That's just - he's standing where the ham station - you know we're going to have a full AM radio station, too. I kind of went overboard. Stop me. Somebody should handcuff me before I spend again. All right, Steve. We've got to do a show.

Steve: Let's do a show.

Leo: So let's get Security Now! underway. All right, Steve. I guess, as always, we need to start with security updates.

Steve: Yes, we do. This is a little more of an old-style Q&A inasmuch as we don't have a ton of news this week. But I've back-loaded us with lots of questions. We've got a full 12 useful and interesting questions. I had time to go through, and some of the earlier ones that I chose I found better ones for them. So I think we've got a really great podcast with a little bit of news and a bunch of miscellaneous stuff. But mostly a Q&A that's actually Q&A this week.

Leo: Well, I love it.

Steve: The real only significant, worth mentioning update I ran across wasn't even PC. It was for iOS. And I'm sure you have already tuned into this, since this was on Friday of last week, that Apple released v4.3.4 of iOS for all of their iOS-based devices on the AT&T platform. The Verizon CDMA customers get updated to 4.2.9. Users will want...

Leo: I was afraid of this kind of forking. That's really too bad.

Steve: I know. Users will want to update, though, because this fixes three critical problems that involve, well, or include a PDF rendering flaw which was announced by a hacker, Comex, as a slick means of jailbreaking iOS devices. Well, it's one thing if a user deliberately loaded a PDF, which is what this is. This is a PDF rendering flaw. Now, this is not something we can blame on Adobe, for a change, because this is Apple's own PDF code. So the moment that Comex introduced this concept of jailbreaking via PDF - which is very convenient for users who want to jailbreak their iOS devices, of course, that's not something that Apple wants people to do - Apple announced that they would quickly be addressing this, so they have.

The risk is, though, that people who don't update could have this PDF flaw leveraged against them. Just by visiting a website or viewing a document which they don't think is malicious, they could have their device taken over. So everyone wants to update, anyone who's using iOS devices. In fact, I tried to do that before the podcast, but it told me it was going to take 30 minutes, and we only had 15 minutes to go before you and I got online. So I scrubbed it after I saw that, and I'll do my updates after we're through.

Leo: I have this bad habit of, as soon as I see an update, even if I'm beginning a show - but I didn't.

Steve: Yeah, and that can bite us when it's a 600MB Mac OS update.

Leo: Takes a while. We got Lion is supposedly coming out tomorrow or the next day.

Steve: Yeah. And I have to say, one thing that Microsoft seems to have done right, maybe just because they've had much more time to do it, is their updates are smaller in size. Apple seems to just replace the whole OS every time they do a major update.

Leo: Well, actually this is something supposedly they will fix in iOS 5.

Steve: Ah.

Leo: Yeah, that they're not - they're going to have delta updates as opposed to - in fact they do give you the full firmware each time.

Steve: Right. So the only thing I had, and I don't really have any information about it, but I got a bunch of people saying, oh, Steve, are you going to talk about the fact that LulzSec, our Lulz Security folks, hacked The Sun newspaper in the U.K. and put up their own content on the site, which was embarrassing for the paper. And so it's like, okay, well, I'll mention that. But...

Leo: Even more embarrassing, frankly, for the British television newscasters, who said, "Well, it looks like The Sun's been redirected to 'We're in it for the Louise.'"

Steve: Exactly. Okay. And I did tweet something earlier in the week that a lot of people retweeted because I thought it was interesting. And this was a - it was picked up by ReadWriteWeb, which had a snapshot of a Forrester research survey that showed that Windows XP still powers 60 percent of corporate desktops. And it also showed Apple making a small gain. I have the link to that...

Leo: It could have been worse. It could have been Windows 98.

Steve: Yeah, it could have been. And so I'll just read a little bit from this. It says, according to a new report from Forrester - and I tried to go there, but they want some thousands of dollars for it. So I thought, well, okay, I'll just cite what ReadWriteWeb had rather than trying to talk Forrester out of a copy. "According to a new report from Forrester, Windows 7 is now in use on 20 percent" - okay, 20 percent only - "of corporate desktops as of March 2011. Windows XP still holds onto 59.9" - rounded to 60 percent - "of the enterprise desktop world, down from 67.5 a year ago." So a year ago it was at 67.5, and it's only dropped to 60 percent over the course of the last year. So only - they only lose 7.5 percent. Apple now has an 11 percent share of the corporate desktop, up from 9.1" a year ago. And Linux, still struggling for much, is only at 1.4 percent, and it was at 1.3 a year ago. So Linux is really not seeing much traction on the...

Leo: Oh, it's growing. It's growing.

Steve: It's in the right - it's going in that direction.

Leo: The right direction.

Steve: Meanwhile, IE, Internet Explorer, has declined slightly, as we've talked about in several opportunities recently, while Chrome and Safari are both on the rise - of course Safari being brought along by the Mac that is also on the rise. It said, "The pace of Windows 7 adoption is accelerating, according to the report. Windows 7 dominates new deployments, with XP and Vista finally starting to disappear. Forrester says Vista adoption peaked in November of '09 at 14 percent and has declined ever since."

So we're seeing that - and this is sort of what we would expect, which is new systems which are probably reluctantly being purchased, just because they're not free, are coming with 7 built in. But corporations are not in any rush at all to just upgrade the OS when they don't have to. So they're staying with XP. And XP is phased out, they're being replaced with Windows 7. And Vista sort of never really got a foothold. It wasn't there long enough for it to replace the XP systems. And 7 came along, and so all new systems being deployed are going to be Windows 7 based rather than Vista. So that's sort of the arc of that. And this link does have an interesting table that shows how all this breaks down over time. So I just wanted to - I caught that and thought that was interesting. We've talked about legacy machines and security and so forth.

There's a really interesting - and Leo, you should go to this URL while we're talking: Collusion.toolness.org.

Leo: Okay.

Steve: Collusion.toolness.org. And this is - and I also tweeted about this, so people could also check my recent tweet stream. And by the way, I am on Google+ as just Steve Gibson. You can search for "Steve Gibson" and find my picture there.

Leo: I found you, I found you.

Steve: And what I'll be doing from now on is duplicating my tweets or my postings over there...

Leo: You might find that the thing about Google+ is it ends up being all about conversations, interesting conversations.

Steve: Right.

Leo: And while you can do that on Twitter, it's a little more difficult to follow.

Steve: Yes. So, okay. So Collusion is actually a Firefox add-on which I have not yet had a chance to take a look at. I should explain a little bit that I was a little caught off guard by switching the podcast from Wednesday to Tuesday. We're recording this a day earlier, and so I had budgeted some time that I will use next week to have a chance to play with Collusion. What this does, it sort of takes us the next step forward from Ghostery to actually tie together sites that are colluding in tracking us. And so this Collusion.toolness.org just sort of gives you a quick overview example. It takes you through several sites and builds this collusion graph that shows how different tracking which is common to those sites links you together as you move from site to site. It's very interesting and sort of interactive. It builds a cool node graph on the fly as you move through, I think it's four or five different sites. And then you are able to hover your mouse over the nodes, and it pops up information about them. You can drag them around actually to sort of reorganize this node graph, which is sort of...

Leo: What's interesting, though, is as you go from site to site, it shows the interrelationship between the sites.

Steve: Yes.

Leo: So it's more than just what this site tells you and what that site tells you, which is what Ghostery tells you. It shows how sites, successive visits to sites build up a picture.

Steve: Exactly. And so, again, I have not had a chance to look at the Collusion add-on. But I'm absolutely, I mean, Ghostery is really cool. And this thing running on Firefox will link all this together. I can't wait to take a look at it and see what it looks like.

Leo: Yeah. This is a really nice visualization of this information. And even if you don't want to install it in Firefox, if you go to that Collusion website, he kind of gives you a demo, which we're running through right now, of how visiting sites will build up this amazing graph of information about you.

Steve: Yeah, yeah. So many people have tweeted me about Mozilla's new identity effort called BrowserID. I wanted to just acknowledge that I am aware of it, and I've seen it. I'm going to give it a full analysis because I dipped in quickly, thinking maybe I could take a quick look at it. And it's like, whoa, okay, this is some serious crypto technology that I want to understand and share with our users. But I'm encouraged because of course we did a whole episode on the issue of identity on the Internet. We recognize the importance of it now and certainly in the future. And Mozilla has an approach which they have put together, it's an outgrowth of a project that they have worked on which is still in its infancy. But they use your email address as you proving ownership of an email address with an email loop in order to solve the authentication problem. So I haven't looked in detail at it, but I imagine it's going to be worth a whole podcast where we pull this thing apart in detail and look at how it works, which we will do.

I also noted that TrueCrypt, that is our favorite whole drive encryption system, is looking for money. They're asking for donations. They want to raise $150,000, and so far they have $10,000. So I just wanted to point that to our listeners, users of TrueCrypt who want to support the free and open source effort. I would encourage people to go over and maybe give them a few bucks.

Leo: Yeah, should be a no-brainer. Everybody uses it. And it's free.

Steve: Yes, exactly, yes. And a spectacularly secure and robust solution. And then, finally, there's an interesting site called BrowserScope.org. So it's just as it sounds, BrowserScope, as in oscilloscope, but this is a browser scope, dot org, that shows and tracks and allows users to run a series of tests against their own browser of choice to see how it does. It takes about four minutes to run the tests. You will want to enable JavaScript, or you might as well just not bother because it's a heavily scripting test. But it performs a wide spectrum of security-related and function-related tests on whatever browser you visit it with and also shows what all of the users of BrowserScope.org collectively and anonymously have found out about their browsers and sort of builds a, you know, these browsers do this, those browsers do that. So it's just another cool way of looking at where your browser ranks and rates in terms of functionality and security and cookie handling and so forth, relative to the cross-section of all browsers that are currently use.

Leo: I'm sorry. I'm just looking at mine. Interesting.

Steve: Yeah, it's very cool.

Leo: Very interesting.

Steve: So I wanted to recommend that. And I have a short note from a happy SpinRite user, David Ward, whose company is Wellmax Computer. He wrote and said, "I own a small IT consulting business, Wellmax Computer. I've been using SpinRite for almost 20 years." So, wow. Thank you, David.

Leo: Wow, yeah.

Steve: He says, "A customer brought an XP machine in that was freezing. I knew that he was having problems with his hard drive. He wanted a new drive, not just to repair the old one. I had a drive cloning utility, shall remain nameless" - for whatever reason - "and when setting up...

Leo: You'll see.

Steve: "And when setting up to clone one drive to the other, I was dismayed to see that it was projected to take 22 hours to finish the cloning process. This was a 500GB drive. Well, after 24 hours it had barely budged, maybe done 8 percent. My gut told me, I think the source drive is having problems. So I ran SpinRite at Level 4. Two days later, SpinRite had finished. I started the cloning process again and this time was done in 22 minutes instead of 22 hours to get 8 percent of the way done. I got to thinking, why don't these cloning software manufacturers include SpinRite as part of their software? Of course there would be a royalty to you for each copy sold. This just seems like a match made in heaven. David Ward."

Leo: That's a good point.

Steve: Well, and I'll just say I really avoid any sort of that kind of thing. I have a very simple business model. I get people sometimes who want to license SpinRite or bundle it with things. And then we're doing tech, I mean, we would still want to do tech support, but then they'd say, oh, no, we'll do tech support, except they can't do tech support because they don't know the product that well. Customers would end up being annoyed. It's just I like things - I like to keep things simple. So I wanted to share David's experience and let anyone know that if they have, like, a problem cloning a drive, to remember SpinRite because it can useful and helpful there, as well. Hey, you're a ham now, and you've just got your ham gear, so...

Leo: You saw that, did you? Yeah.

Steve: You'll be a little distracted, I think.

Leo: I passed the technician test on Friday. And I want to thank everybody at the Mount Diablo Amateur Radio Club - many of whom are fans, by the way, Steve - for being so nice to me. And, you know, it was just - it really felt good. I haven't taken a test for anything in years, so I was a little nervous. And then, now, soon I shall get my call letters. I don't know what they're going to be yet. It should happen in the next few days. But I have to admit, part of the reason I became a ham is for the gear. And I've already, you're right, I've already purchased my radio. I'm ready. The minute I can. I can't yet. But the minute I can, I'm going to pull this radio out of its cup holder, and I'm going to say hello. Actually I want to learn Morse code. Did - you never became a ham.

Steve: Never became a ham. I had to learn Morse code for the Boy Scouts because that was one of the merit badges that you earned for something, I don't remember now what. It may have been a communication - I remember we had semaphore and Morse code and screaming really loud and various things.

Leo: That's - so I'm jealous because that's the next thing. Although I'm looking forward to learning it, actually. I have a - this is where software really can come in handy. There's a Morse code trainer I've downloaded for Windows, and it's amazing. And I think I'll learn much more quickly than I would have learned if I were trying to piece it out by myself. So this is fun. It's a good little hobby. But enough of that. You ready for some Q&A?

Steve: Because you don't have enough hobbies, Leo. You're just - you're casting about for something to do all the time.

Leo: I actually don't really have a hobby. This is my profession. Fortunately, my hobby is my profession. I love technology. And so when I go home and play with Ruby or read tech news, I mean, it never feels like work to me. It's all I do, all the time. Even when I'm building my tiny tower, I feel like that's a little bit of work. Before we get too far down that road, let's get some questions.

Steve: Good idea.

Leo: Steve Gibson, Q&A #122, starting with Joey, a high school student listener in Alberta, Canada. He wants some career advice: Dear Steve, I just recently started listening to your podcast, and I have to start off by saying you're doing a great job. I've slowly been working my way through the extensive list of archived episodes. I've been learning a lot. Thank you. I'm a high school student with a passion for computers and math - sounds like, this kid sounds like he might be a little mini Steve, actually - and I want to become a computer security expert once I graduate because that's what I'm most interested in.

However, with the constant evolution and improvement of computer security and crypto, I'm worried that by the time I get through high school and university there'll be no more improvements to be made. Oh, fear not. So my question to you is this: Do you think the computer security industry will be around long enough to sustain a lifelong career for someone my age, or will it soon become obsolete? How long will it be before every end-user machine becomes secure enough so as not to require constant attention? I'm guessing that will be a long time, but I want your opinion. Any advice is appreciated. I'm currently at that point in my life where I need to focus on a career path and decide which computer-related field I'll specialize in. Computer security is my first choice, but I'm unsure how viable that career will be in 10, 20, or 30 years. By the way, I'm currently working toward my A+ - this is a high school kid, by the way.

Steve: Yeah.

Leo: What did he say, he's a junior?

Steve: I don't think he says.

Leo: He doesn't say, but he's in high school. Currently working towards my A+ certification and plan on getting a BS in Computer Science focusing on Information Security. One last question: Would SpinRite work with my PlayStation 3, iPod Classic, or Shaw DVR system? All of them use hard drives, but are proprietary in nature. I plan on buying SpinRite in the future to show my support, but want to know if these machines are supported/tested. Joey, Alberta, Canada.

Steve: Well, okay. So, first, just so I don't forget, yes, SpinRite will work on all of those.

People have fixed - we're heard many reports of PlayStations being fixed with SpinRite. We told some stories years ago about the classic iPods. You may remember, Leo, there was one guy who was collecting them because he sort of became the guy that collected dead iPods, the original ones with hard drives in them.

Leo: Yeah. That's a business for this guy.

Steve: Yeah, all of his friends were just giving him their dead iPods. Then he discovered SpinRite and began running SpinRite on his iPods and fixed them all and started handing them back to the friends saying, hey, here's your iPod, it works again. So, yes, it absolutely will work on non-standard anythings. You have to disconnect it and preferably hook it to a PC, that is, take the drive and hook it to a PC. But if you do that, SpinRite will say, oh, there's something spinning here, let's make it work.

Leo: It doesn't need to know anything about the file system.

Steve: Correct.

Leo: The only issue would be, on the Shaw, if it's an encrypted file system. No, I guess it doesn't matter either...

Steve: Doesn't matter.

Leo: ...because it's all underneath that; right?

Steve: Yup. And in fact the TiVo uses a byte-swapped Linux because the original TiVo was a PowerPC chip that used a big-endian rather than little-endian byte ordering, which was the byte ordering of the PowerPC. And SpinRite says, I don't care. It just fixes it.

Leo: That's awesome.

Steve: So whatever.

Leo: Well done.

Steve: Okay. So is security going to go away? There are a couple things, a couple reactions. First of all, when people generically ask me what they should do, my reaction is always, the way you want to spend your life is doing what you love. So if you love computer science and security, Joey, as you say you do, then I would worry less about what you'll be doing in 30 years than what you'll be doing in 10 years. And we know that these problems are not going to be solved immediately. We'll discuss the future in a second. But there's just - I would never suggest you do something that you don't love because you're worried about 30 years from now than doing something that you do love for the next 10 years. Or probably for the rest of your life. So there are so many high school kids who don't love anything. I mean, they love videogames, and they love hanging out, but...

Leo: They love their girlfriends.

Steve: And the girlfriends, but they can't get a job doing that.

Leo: Right.

Steve: So the idea - and to that extent, Leo's right. You are a mini me inasmuch as, just like Leo, who loves what he does, when I talk about, oh, I'm going to get a lot of work done today, it's because I don't have any distractions scheduled, and I just get to do this, computer stuff, all day long. There's nothing I want to do more. So, boy, if you can spend your life doing what you love, I mean, and having someone pay you for it, there's just no better way to live.

Now, as for is there a future in this, I've often said that some of this feels to me like the Wild West, like we're still in the frontier era. And I joke about how this just can't stay this bad forever. Except it seems to keep getting worse rather than getting better. I mean, we're seeing improvements, like Microsoft is finally not executing scripts in email by default, as they were during the beginning of this podcast. That's been fixed. But boy, we sure don't see any slowdown. I mean, we've added a new section of breaches and break-ins because there's just been such a rash of that.

So I see things becoming - evolving. The threats seem to be getting more sophisticated and requiring a much higher level of expertise to deal with them, rather than being simple. So to me that says there would be more specialization in security, but certainly no obsolescence of that. And even if, in 10, 15, 20 years the challenge changed, you'll have a decade of experience in computers and technology and knowledge and detail stuff, and that'll evolve. I mean, I started off being Mr. Hard Drive, and my focus has shifted to security because there was a need and an interest.

So nothing prevents you from having your own expertise evolve over time. And growing and learning is really how you want to spend your life. And the creation of the Internet completely changed my focus. So who knows what your focus will be 30 years from now. But I would say do what you love; and, if that changes over time, well, that's growth, and that's a good thing.

Leo: I think it's fairly safe to say that it's only going to get worse.

Steve: Yeah.

Leo: I don't see how security will ever not be a huge issue.

Steve: Well, and, yes. Another maybe great analogy is to look at whether security in the physical world, which is mature as people are...

Leo: Right, right.

Steve: ...that hasn't gone away. I mean, the whole security issue, even non-computer security is, I mean, that's why we have the word "security." The word came along before computers did.

Leo: Actually physical security is an interesting field. I wouldn't even eschew that. Even just locks. I went to a talk on locks and lock-picking. It's fascinating. And there's a huge resurgence in trying to make physical security systems that work. So you couldn't go wrong. I think there are two areas that in computer science are guaranteed to give you employment for the next 40 years. One is networking, and one is security. And they're kind of related.

Steve: Yup.

Leo: Listen to this show, you'll know all you need to know. Good question. Good luck. Jimmy Blake, Sandusky, Ohio wonders whether we're trying to mess with his head. He said that. You've probably received this from a number of people already. But as someone who has recently decided to go back and listen to your earlier podcasts in the Security Now! series, after realizing how in-depth you get with topics, it threw me for a loop when I saw "How the Internet Works," both as the newest episode for me to watch last week, after having just listened to episodes 25 and 26. Which apparently, I didn't know, it has been a few years, were "How the Internet Works."

Now, that's not to say you shouldn't be revisiting the series because I love the new depth you're going into. And I believe that having several years of podcast experience now under your belt you're doing a more thorough job explaining the principles of the Internet than the first time around. I presume, Steve, that this is not a surprise to you. You knew we did it once before; right?

Steve: Yup.

Leo: Yeah. Thanks for all the hard work you do in always giving me a little more information on the topics I thought I understood pretty well. P.S.: This is way off that topic, but after listening to some of your older crypto episodes and watching the more recent episodes on randomness, I was wondering about using a hashing or an encryption algorithm as a source of random numbers. If, for example, you were writing a program in a programming language with poor random number generation (*cough *BASIC* cough*), but that also did - actually they all have lousy random number generation, that's another matter entirely - but that also did have built-in functions for hashing or encryption, couldn't you generate a random string using the built-in random number generator of the programming language, then feed that string through one of the hashing/encryption functions and then use the raw binary of the result as something with better entropy than the built-in random number generator could produce on its own, since hashing and encryption algorithms generally rely on a lot of very good randomness? What about it, Steverino?

Steve: Okay. So two things. I did know, of course, that we did a series on how the Internet works five years ago.

Leo: Whew. I forgot.

Steve: Episodes 25 and 26. And I decided it was worth revisiting, deliberately, because for one thing I know that we've got a huge bunch of listeners who are now tuned in and listening and may not have gone back there. But also because I'm taking sort of a different approach. What I said back on Episodes 25 and 26 is not what I said last week. And, by the way, I've just had a ton of really great feedback from last week's episode, our revisiting how the Internet works, which I think is worth doing every five years or so. So I'm going to, even though it's got the same or similar title, I'm going to bring a fresh approach to our looking at it again. And I'm sure that people who've even been listening from the beginning will get more out of it in hearing new stuff about how the Internet works this time. So, yes.

Leo: It's not like the technology's changing.

Steve: The technology is not changing, although this time of course we will talk about IPv6 and its influence on how the Internet works. So some things are changing. DNS spoofing has happened since then. And so I'll pull everything that's happened then into this revisiting that. But also I'm going to do it with a little more style, maybe a little more in-depth this time, so that it builds on everything that we have covered in the last five and a half years. It gives me a better foundation for talking about new stuff in additional detail.

As for random number generating in, like, languages that have bad ones, the problem with the random number generators in most languages is not the algorithms so much as that they lack a good source of entropy, that is, a good source of seeding randomness which feeds the random number generator. So if you just switched it over to a crypto-based approach, which potentially can generate very good random numbers, you still have the problem of, if your intention is to make it unpredictable, then you need to somehow arrange to give it entropy in order for it to churn out random numbers based on a good algorithm, but from an unpredictable starting source. So that's really the trick.

I solved it with various of my recent pages where I've needed to generate randomness. And we talked about one, that R&D page, I think it's GRC.com/r&d/js.htm - "js" as in JavaScript dot htm - which is where I wanted to develop the technology to do that for some future projects. I have a 256-bit token which is received with that page from GRC, which I then hash with a whole bunch of client-side stuff in order to generate something which we know has the entropy available from GRC, but then we then further scramble it so you're not actually taking what GRC's server provided. You're adding entropy to it over on the client side, which really gives you the best of both worlds. You know you'll have at least as much entropy as you got from us, which is a lot of entropy because we've got a very good random number generator running on the server at GRC, yet you won't have the security problem of relying on something from GRC. So all you're doing is you're adding to the entropy you received, which guarantees you have security from GRC knowing what you're using, yet you have the added entropy that you're generating on the client side. So it's possible to do clever things like that.

There is a site, I think it's Randomness.org or RandomNumbers.org, there's some - it's been around for a long time, for years, a source of entropy that anyone can pull on the Internet, that you could use from a basic program, for example, to generate some seed entropy, and then maybe add to it by following the mouse around or looking at the time of day on the local machine, just pouring all that additional, hashing all that into a seed which you would then use to start off a good cryptographical-based random number generator. So those problems can be solved, and they're fun to solve.

Leo: We have a tweet from @greendrive, Allan Hoiberg in Denmark. He's making a very good point when he asks: What's to prevent IsMyCreditCardStolen.com - by the way, we talked about that last week, and I posted that on Twitter, or maybe it was Google+, and boy it got a lot of scared people. But he said, what's to prevent IsMyCreditCardStolen.com, which was a site created by a very reputable group, the Anti-Phishing Working Group [Antiphishing.org], from serving up a page that does actually post data once in a while? In other words, could it have been malicious? By the way, I don't know if you know, Steve, but OpenDNS blocked it as malicious.

Steve: Yes.

Leo: They didn't bother to look deeper, obviously.

Steve: This was - I loved this question because it's something I failed to address. And it comes up whenever we're talking about something that looks trustworthy, or for example I mentioned that I looked at before, I mean, when I found the site, and it's saying put in your credit card and your name and your billing address and your expiration date and things, it's like, no. And I never did, but I looked at the source, and I saw what it was doing.

So Allan makes the point, which I wanted to reiterate or really remind our listeners, and that is what a site is doing today isn't necessarily what it's going to be doing tomorrow. And that's a super valuable lesson. So you're right, Leo, this is from really good people. You'd want to make sure that it was being delivered over HTTPS to prevent a man-in-the-middle attack from changing what that page is delivering. And this is related to what we talked about with Google, which was that Chrome is now, if your page comes in over HTTPS, the Chrome browser is now enforcing and insisting that any scripting for the page also come over HTTPS, that is, not allowing sort of the laxity of a mixed content where the scripting would not be as secure as the page delivery. Other browsers would warn you that there was mixed content, but wouldn't absolutely refuse, for example, to run the script if you said, oh, yeah, go ahead. Whereas Google Chrome does, which is a good thing.

But it is a lesson to remember, that any site which is behaving in a trustworthy fashion on one day, which you determine for example by looking at what the scripting is, isn't necessarily doing it on some other day in the future. So it's worth keeping your guard up all the time. And I just - that caught me off guard. It's like, yep, you know, Allan's right. We cannot assume what's going to happen in the future. We just don't know.

Leo: I don't - was it irresponsible of me to repost that link?

Steve: I don't think so because it's from good people. I would say the benefit far outweighs the liability.

Leo: Certainly raised awareness, let's put it that way.

Steve: Yes, yes.

Leo: Tim in Maryland wants to know more about randomness. He says: Steve, you talked about how randomness is a very important cryptography issue, but I just don't understand why. How can you use something random in a process that must be repeatable in order to decrypt? I thought the whole idea is that your password is somehow used to generate a key, which is then used to encrypt the data. You certainly can't use a random key because how would you ever decrypt it? This has bugged me for some time now, and I can't seem to find any clear description on it. I'm sure it matters, but I don't know why.

Steve: Okay. So it's a great question. The best example and the cleanest example is the use of randomness, which is very common. For example, every time we make an SSL connection, or also known as TLS, Transport Layer Security, and which we see as HTTPS: on our browsers every time we're putting credit card information or logging in securely to a website, it's establishing a secure connection. And randomness plays in that. The idea is that you use public key encryption, that is to say, asymmetric encryption, to establish an encrypted connection between the two points. But the asymmetric encryption is incredibly slow. You cannot possibly afford to use asymmetric encryption where the server, for example, gives you its public key. If you use the public key to encrypt your content, you could send it to the server, and only the private key could decrypt what you encrypted with its public key. So that's feasible, theoretically.

But the problem is that public key crypto, which is what we want to use because of its asymmetry, that whole power of having a public key and a private key, we want to use it, but it's just too slow. So this is where randomness comes in because we do have an encryption technology which is lightning speed, and that's symmetric key encryption, where we use the same key to encrypt as we decrypt. Except in this case we want the effect of public key encryption without the computational overhead.

So what happens is, and I'll just take a very simplified model to keep this clear, imagine that instead our side, our sender, we get the public key from the server. Now we make up, we use randomness to create a short, 256-bit for example, symmetric key. That we encrypt, we encrypt that short symmetric key with the server's public key. So we're only encrypting something small, one time, with the public key technology. And we send that to the server, which only it is able to decrypt using its private key. And now I know what the symmetric key was. The server has our decrypted symmetric key. And now we can transact back and forth at very high speed and with low computational overhead using the symmetric key.

So the point is, in order to make that secure, we have to be able to generate a symmetric key randomly with very high quality so that no bad guys can guess what that symmetric key is. So there's a very clean instance, or a simplified example for how randomness is required for cryptography because it's just far easier for us to use asymmetric encryption to create a short key which is then transmitted to someone who can decrypt it. And then we use that short key for the actual bulk payload transaction. And it works.

Leo: I always liked that kind of clever solution. It's the same problem you had in the old days, and this is how I remember it, when you wanted to encode a message. You'd have to send a key somehow. And that was a huge security flaw because you have a symmetric key. It's the same key used to encode as to decode. So public cryptography kind of broke through that problem.

Steve: Absolutely.

Leo: But it still had the asymmetry of computational difficulty. Greg in North Hollywood, California wonders about memory hard functions: Steve and Leo, in Episode 296, Listener Feedback #115, you said, quote, Steve, "But imagine an algorithm which is memory intensive, that is, where in order for it to function - and there's just no way around this - it has to be given a big, like a gig, memory block. It has to have it all to itself for some length of time in order for it to do its job. And there just isn't - you can demonstrate cryptographically - there's no way to do this without it having access to all of that memory. And those are called 'memory hard' problems."

So he continues: I'm running a desktop computer with 8GB of RAM. I need it for VMware - long story. Some desktop motherboards support up to 32GB of RAM. As you noted, GPUs, the Graphics Processing Units, have 4 to 6GB of video RAM. Some motherboards will support up to four video cards. Now, I think that the context was using FPGAs, Field Programmable Gate Arrays - I'm not following this at all - and the problem of an FPGA having a few gigs of RAM. Is that the problem? Steve. I don't know what he's talking about.

Steve: So it was a great question.

Leo: Good. What does it mean? Would you rephrase as a question?

Steve: What he's talking - yeah. And it's something that we will cover in more detail in the future. But we did talk about it, as he mentioned, back in Episode 296. The idea was that massive password-cracking systems are able to, for example, test a huge number of hashes in a short time because the hashes were designed to be very efficient, like an SHA-256 is deliberately designed to be very fast. So, for example, one of the ways that we increase the security is we make you do the hash many times. The WPA specification requires 4,096 repeated hashes of, like, you just hash, and then you rehash that, and you rehash that, and you rehash it in a loop 4,096 times. So however much time it takes to do one hash, this at least makes that 4,096 times longer.

And due to the nature of hashing, no one knows how to short-circuit that. No one knows how to just, like, do one operation that gives you the result of 4,096 hashing operations. Because it's cryptographically strong, we don't know how to do that faster, forcing anyone who wants to generate a trial key, for example in the case of WPA WiFi encryption, they're forced to do this 4,096 times.

Well, smart people have said you could design an FPGA, a Field Programmable Gate Array, because they're getting bigger and more powerful, where you built 4,096 SHA-256 hashing engines on a single chip. So essentially you move this problem out of the time domain and out of, like, a loop which our computers have to execute, just into silicon, where you put a test in, and this piece of hardware just goes [sound], and just in no time at all, almost like in a big pipeline, where all 4,096 of these SHA-256 hashing engines are going at once, processing hashes, you can make it extremely fast.

So because of the threat represented by our ability to create algorithmically dense silicon, the leading edge smart guys who think about how do we increase security, they've said, wait a minute. It's one thing to have algorithmic density. But if we also required memory density, that is, if we required, if we came up with some sort of clever functions which are what they call "memory hard functions," that is, functions which are not algorithmically dense, but they're memory intensive, then there's no way, for example, you could put 4,096 of those, if each one of them required a gig of their own, well, that would be 4,096GB in order to do this same process.

So essentially this is sort of a theoretical construction at this point. This is the leading edge of where crypto theory is going in terms of, okay, we're seeing what GPUs are doing, we're seeing what field programmable gate arrays are allowing. How can we keep the security of cryptography high? How can we keep that bar high in the face of algorithmic speed? And what they're seeing is that requiring large blocks of memory, which our contemporary machines do have, but which screamingly fast algorithmic density doesn't yet have, is one trick we can use for keeping these things hard and slow. So that's what that was about.

Leo: Well, I'm relieved to know. Randy Hammock in Sun Valley, California has an AC power line frequency tale: I had an uncle who worked in a small power plant. We were talking last week, I think, about how the fact that the 60Hz electrical power was going to kind of be deregulated so it wasn't exactly whatever it was, 59.98Hz or whatever. I have an uncle who worked in a small power plant. Their frequency regulator was a clock with two second hands. One second hand was driven by the power line, while the other was driven by a very accurate pendulum clock. Oh, boy.

The pendulum clock speed was controlled by adding and subtracting small gold weights on a balance scale. WWV was used to verify the pendulum clock time. That's the world clock in Boulder, Colorado, the radio clock. Once the pendulum clock was calibrated using WWV, the generator speed would be adjusted to keep the two second hands in sync. While the short-term frequency of the generator may be awful, this system would pretty much assure a two-second-per-year accuracy, which was what the government required. A really neat example of a generator being manually phase-locked to a highly accurate standard. That's a cool way of doing it.

Steve: I just thought that was cool. I mean, you can imagine, I mean, I guess it was a small power plant. And I don't know how long ago this was. But the idea of going into the power plant, and here's some wacky clock where you've got two second hands, and the engineer somewhere is looking at them, literally seeing which one is ahead of the other and speeding up or slowing down the generator whose actual spin, the actual coils on the generator are what's producing the sine wave output that this power plant is putting out onto the grid.

And so they're tuning the speed of the generator which is directly driving one of the hands of the two second hands on the clock, the other one being driven by a pendulum. And pendulums actually keep really good speed. You need to also keep them constant temperature or to have them temperature compensated because, if the pendulum is made of metal, we know that metal expands when it gets warmer, and that would tend to slow the clock speed down. But then they're also backing that up with WWV, which is the Fort Collins, Colorado time standard, as you mentioned, Leo. So overall it looks like it's pretty good technology.

Leo: I said Boulder. Fort Collins, of course.

Steve: Fort Collins.

Leo: Drew in Atlanta wonders about the value of IP filtering: Steve, my Google account was recently breached.

Steve: Oops.

Leo: Whoops. I was made aware of the breach by Google reporting unusual access to my account from a Chinese IP address. I have never been to China. Luckily, I was able to update my password, and I haven't seen anything unusual since. But it still leaves me a little uncomfortable with the idea that my account could be accessed from anywhere when I usually only find myself in a few specific locations. So I was wondering, how practical would it be for a company like Google to set up user-controlled IP filtering for individual accounts? As a network admin I frequently use IP filtering to prevent malicious use. It would be nice if I could go into my account security settings and explicitly tell Google my account should never be accessed from outside the U.S., or drill down even further and say my account should never be accessed from outside the subnets used by my home ISP and smartphone carrier.

Anyway, it was just a thought I had. I was curious to get your take on it. I've been working in IT for more than six years now and have been listening to Security Now! for nearly the entire time. Interestingly enough, my very first job in IT was to rebuild computers compromised by the worms you discussed in Episode 1. Thanks for your time, Drew. Yeah, our sysadmins do that for my server, anyway. I know because I was trying to SSH into my server, and I said, I can't get in. And they said, oh, yeah, we hadn't added that IP address. So that's great. It's a good way to lock it down.

Steve: It is a great way to lock it down. And I, similarly, use that for accessing non-public services at GRC. I've got GRC's network is remotely located in the Level 3 datacenter in nearby Tustin. And there are - I use IP filtering to the fixed block of IPs that I have here as a further way of authenticating myself to my system. I certainly don't rely on that exclusively. But it's just one more useful filter. And as we know - and we'll be talking about TCP protocol before long in our How the Internet Works series. And I've often said that TCP, unlike, for example, ICMP or UDP, is non-spoofable, that is, you cannot spoof the source IP because in order to set up a roundtrip communication, the far end has to send packets back to you, so you've got to be available at that IP.

I would say to, like, in terms of how can this be fooled, that it is not absolutely impossible to fool IP filtering, though, because, for example, say that you were to lock down filtering so that you could deal with, for example, the fact that your ISP might vary your IP address. Drew understood that because he referred to "outside the subnets used by his home ISP and smartphone carrier." Meaning that he recognized that the actual IP might change, but the network that was being used would probably not. So it would stay within a range of IP addresses.

The problem is, a proxy is something that bounces traffic through itself so that, if you were a high-value target, a bad guy could compromise some other machine within your ISP's network and then access your Google account, for example. So if Google were told to accept IPs within this range in order to allow you the flexibility of coming in from any IP address within your ISP's range, then a bad guy could get in using a compromised machine within that same IP range. So there are ways around it. I mean, overall I like the idea. My sense is it's probably a little too tech-y for Google to do, and not the kind of things that their users would be able to do without getting themselves in trouble, or needing to then fall back to a higher level of authentication.

We were talking about this in the context of the World of Warcraft and the Blizzard authentication token and how they're using apparently some IP filtering or IP identification and some tokens on the machine in order to identify their users. Your sysadmins, I'm glad to hear, Leo, are doing that kind of thing. It's another good level of verification. But a perfect example is that you got caught out by it. So it is something that comes with some cost from an administration standpoint.

Leo: You know, Google, if you go to the bottom of your Gmail page, does give you information. I mean, you can look at it, if I click the Details link at the bottom of my Gmail page...

Steve: The most recent logins.

Leo: ...it says "activity on this account," and it gives IP addresses of all the most recent logins. And this is very, very useful. And if you see China on here, and you don't go to China, you might want to wonder. That's what Google's doing lately. But this is fantastic.

Steve: And I do love it. We talked about this a long time ago when Google added this service, the idea that they would say, whoa, by the way, you should know. And that's just - that's fantastic.

Leo: And of course they have a setting that says "show an alert for unusual activity." So I would say this is about as close as they're going to get to blocking individual IP addresses or only allowing certain IP addresses. It's going to tell you when it sees something that it's not seen before. I think that's pretty good.

Steve: Yeah. And I would like it if, for example, if you had a setting of saying, rather than giving me an alert if somebody else has accessed my account from China, require any unexpected access some higher level of authentication.

Leo: There you go. What's the password? Yeah. Ask me again.

Steve: Yeah.

Leo: Send a message to my cell phone.

Steve: Exactly. Require that exact authentication for those instances, but not normally.

Leo: Right. That's not unusual. A lot of systems do that. Rob Galanti in New York, New York asks about OAuth as a security risk: Steve, I'm a long-time listener in the city. Thanks for the great podcast. While listening to Episode 308, you and Leo were both proponents of OAuth and HAuth. However, I've always thought that using my Gmail credentials for non-Google sites might be more likely to compromise my security. Here's how: A nefarious website posing as a legitimate one requires my OAuth credentials for login - his Google password. Naively, I enter my credentials. Well, I mean, this could happen to you without OAuth. This is called "phishing." Naively I enter my credentials, specifying they're for Google, and submit the form. Upon submit, the credentials are captured by the nefarious website and can be used to get to my Google data and any other site which might use those credentials. Is that a realistic liability? And, if so, doesn't that outweigh the potential benefits of using OAuth or similar technologies? Thanks again, keep the Security Now! podcasts coming. Rob.

Steve: Well, this was a good question because I wanted to make sure people understood, and specifically Rob and anybody else, that's not the way OAuth works.

Leo: Right. I mean, you could be phished; right?

Steve: Yes.

Leo: But so that's - but that's not an OAuth vulnerability, that's a vulnerability, where somebody puts up a page that you think is real and says, hey, login here, and you go, okay.

Steve: Yup. So if a bad site were to say, hey, we support Google's authentication, give me your Google credentials, well, unfortunately some users are going to fall for that. They're going to put their Google stuff in thinking, hey, that's cool, I'll use Google to authenticate because they may have done that elsewhere or know that Google offers that capability. But that's, as you said, Leo, that's not OAuth.

The way a legitimate site uses OpenAuth correctly is they say you can create an account with us, or login with our credentials, or you can log in with us using your Google credentials through OAuth. What happens then is you are bounced over to Google, and you'll see an SSL connection. And so you give Google your Google credentials, and Google authenticates your login and then bounces you back to the site that is using OAuth and Google. And there are cryptographic token exchange sort of behind the scenes that you don't see, that allows this site to say what it's requesting from Google. Google shows you what the site is requesting. You give Google permission to give the site those things that it has asked for, and then it takes you back. So no one other than Google ever sees your Google credentials.

As you said, Leo, it's a little worrisome because it does require that the user understand this. And the very fact that Rob asked the question demonstrates that it's subject to some confusion. And no doubt we'll see people abusing this as it becomes more popular. But it's not a function or a fault of OpenAuth, it's that social engineering attacks are unfortunately successful.

Leo: Yes. I mean, that's the whole point to OAuth was that you would only - you would never give credentials to anybody but the appropriate site, and that they'd pass along a token.

Steve: Right. And I love it. And the early problems, there were a couple implementation problems that got fixed early on. So I really promote this. I'm seeing more use of PayPal. And much as I really dislike the PayPal company, I sure would rather authenticate with PayPal than give my credit card information to some random company that I'm never going to use again. So this kind of thing where, boy, is it slow to happen, but it's a good trend.

Leo: Konstantin in Toronto asks you about the Portable Dog Killer 2.0: Steve, I'm a Security Now! listener since day one. I love the show. I've subscribed, and I listen to all the other shows. But for me, Security Now! hits my sweet spot, baby. One of my favorite episodes is the Portable Dog Killer. And while I enjoyed the philosophical, educational, and humorous aspect of it, I could not see an immediate use of it when I listened back in May 2010.

Well, that is, until now. We bought our first house last September, moving from an apartment. Our neighbor has a very loud - it's okay, Ozzy, he's not talking about you - and sometimes vicious dog - that's not you, no, it isn't, Ozzy - that scares my two- and five-year-old kids when they play outside in our fenced backyard. Ozzy's taking this very personally. I bought the Yard Gard Electronic Pest Repeller from UrbanNatureStore.ca, but it seems to be too weak to chase away the beast. I then found one person, apparently a Security Now! listener, who made the PDK 1.5 and published his progress at Damage.hackhut.com. However, his instructions are not complete and difficult to digest.

You know where I'm going with this. Steve, Steve, Steve, the community needs you to built PDK 2.0 and publish up-to-date instructions on how to make one of my own. I feel it will be against the spirit of invention and exploration you and Leo are trying to inject into the minds of a young generation, but I hope you read this and somehow reflect on one of the next episodes of Security Now!. Sincerely, Konstantin, Toronto, Ontario, Canada.

Steve: Okay.

Leo: I love it.

Steve: So, okay. So I've got to update our listeners on what's going on.

Leo: Okay.

Steve: There's been a huge amount of interest in PDK 2.0. But the first thing we have to do is change the name because...

Leo: Yeah, that scares people.

Steve: ...I'm so self-conscious of the name. So...

Leo: But the PDD, the Portable Dog Distracter?

Steve: I don't think - well, okay. So here's what's going on. The moment, literally the week I mentioned this bad problem that my best buddy has with this ridiculously obnoxious dog next door, the problem went away. So, I mean, he'd been putting up with it for a year. It had been escalating. He'd been complaining to the neighbor. They finally, like, visited or heard the dog when it was doing this and realized the problem, and they've been far better about it ever since. At the same time, there's a serious problem on the outdoor patio at Starbucks in the morning, not with a dog but with blackbirds or crows.

Leo: Oh, those crows, yeah, no, they're crows, yeah.

Steve: Oh, it is like the social center of crow land.

Leo: No, when the crows move in, all the other birds move out. They're the terrors of the neighborhood.

Steve: Well, and they're smart and loud and obnoxious.

Leo: Yeah. And aggressive.

Steve: Yes. And in fact today, I'm not making this up...

Leo: Nobody's going to complain if you make a Portable Crow Killer.

Steve: Well, so we've got to kill, we've got to remove the "K" from the name.

Leo: How about "stun"?

Steve: Well, my friend wants to call this BFG.

Leo: What's that? Oh, never mind, I know what that is.

Steve: And the listeners who should know, do know. So, okay.

Leo: BFG.

Steve: So the problem is that crows' hearing does not nearly go up to ultrasonic, to out of our hearing range.

Leo: No, they're deaf because they're cawing all the time.

Steve: Their hearing tends to roll off around 8 or 9 KHz.

Leo: How do you know this? Have you studied crow physiology?

Steve: Leo, I am building a - I am building something that will deal with this crow problem. So, yes, I know all about bird frequency ranges now, of hearing.

Leo: Oh, Steve. Oh, Steve. Oh, Steve.

Steve: So I'm selecting the pieces. Something will be shared with the world. It'll be microprocessor based. I've found the various pieces. Now, okay. So that's one thread. In the process, I needed to find a good parabolic reflector. Remember, our listeners may remember that I was thinking of using a tuned tube with a tweeter at the end where the tube would be tuned to be a multiple of the wavelength of the sound. I can't use a tuned tube if I want to also be down at a frequency that the birds could hear because, if we did something that was up at the end of human hearing range, the dogs could hear it, but the birds would be unaffected by it. And frankly, for me, dogs are not a problem anymore.

Leo: Now it's those darn birds.

Steve: But they are a problem for many people because the other thing that happened is many people have, since the PDK episode, have sent me links to a range of things. And you may remember when I was talking about v2 of the Portable Dog Killer, I mentioned I had purchased, like, I don't know, 10 or 11 of these things and given them to Mark to try, before finally thinking, okay, I'm just going to have to recreate something that's serious. Okay. So www.amazing1.com is a very cool science-related site that I found when I was looking for some pieces for v2. So...

Leo: Oh, we've got all kinds of stuff. Look at this. This is great.

Steve: Yeah, it's a fantastic site.

Leo: Oh, man.

Steve: So I wanted to first turn our listeners on to it. They've got Tesla coils and Van de Graaff generators and Jacob's ladders, and just it's cool stuff.

Leo: This is great. Hydrogen-powered cannon?

Steve: Uh-huh. So there's a section on there, on that home page, Ultrasonic and Infrasonic Devices.

Leo: Apparently this is not an unusual desire.

Steve: The top one, there is the second thing listed is Animal and Dog Control, but use Property Protection, Ultrasonics for Property Protection.

Leo: Oh, look at that parabolic sucker.

Steve: Actually that's going to be the one I use. That silver parabolic will be...

Leo: It says: "Hear the incredible world of high frequency sounds beyond that detectable." So this is a receiver, but I guess it could be a transmitter.

Steve: Exactly. I've got two of those on order right now, two of those...

Leo: They're not cheap, dude. These are expensive. Couple hundred bucks.

Steve: All I'm doing is getting the parabolic dish.

Leo: Dish. That's only $50, okay.

Steve: Yeah.

Leo: Look at this, a Sonic Nausea Device.

Steve: Oh, keep going down, Leo. There is some serious technology here for dealing with dogs.

Leo: And this stuff is not illegal? Great little electronic device you can use to clear out those guests that never want to leave, long after the party is over.

Steve: There's something that makes you nauseous, apparently.

Leo: High-pitched sounds that are hard to locate. This is obviously into teenagers. Then there's the Phasor Pain Field Blaster, complex sonic shockwaves. The Phasor Blast Wave Pistol. See, Steve, I thought you were a reprobate. These guys are far worse. Phasor Pain Field Generator.

Steve: Yes.

Leo: Intended for law enforcement personnel.

Steve: They have some things that generate 130dB at several meters. And they have grids of high-frequency generators that work in some sort of phased array for dealing with this. So I did want to let listeners know that - because I have no timeline for this. I'm pulling pieces together. I will document everything I do. We'll make it a community project. I found the microprocessor...

Leo: Wait a minute. Ultrasonic Blaster. Sonic shockwaves blow holes in metals.

Steve: That ought to do the job.

Leo: I don't know if you really should be selling this stuff.

Steve: There's something that cavitates water. So you aim it at water, and it makes bubbles form in the water. So...

Leo: This has to be - this has to cause cancer. That is just unbelievable that this exists.

Steve: So, yeah. I wanted to let people know that there are technologies around, there are devices, not super inexpensive, not like your little handheld thing that does nothing, but these things really do look like they work. So...

Leo: They also have all sorts of insect electrocutors, dog and rodent control devices. Here's a sonic bird chaser might just work fine for you.

Steve: So it's Amazing1.com. And that particular page is /ultra.htm, for reasons we now understand.

Leo: Well, this will do. Canine controller, it's got four tweeters. Holy cow. And a pain field burst section. Whoever thinks this stuff up has got a problem, I think.

Steve: Well, I mean, they've got some Tesla coils that are larger than people.

Leo: Yeah, huge.

Steve: I mean, huge. It's a cool site. I thought our listeners would get a kick out of the site.

Leo: Oh, it's so cool.

Steve: And it will solve the problem for people who are - who want a problem now.

Leo: Got ten grand? How about a two-million-volt output generator? Tesla coil? Why not?

Steve: So my gadget is going to be very flexible. For example, the plan is you'll be able to aim it at a crystal glass and twing it with your finger. This thing will record and lock onto the frequency, and then you'll be able to turn it around and destroy the glass by shooting it with the same frequency that is its natural resonant frequency. So anyway, it's going to be a modern, microprocessor-based, next-generation toy that I'm going to put together. And it'll be all open source, and I'll share all the pieces and parts and everything with everybody.

Leo: We have a little kitty cat lover in the chatroom, says is it safe for cats? I would imagine, if it scares dogs or birds, cats aren't going to be happy about it, either.

Steve: Yeah, they're not going to be happy.

Leo: No.

Steve: Yeah. So I don't know what "safe" means. This is, I mean...

Leo: Unsafe.

Steve: That's why we're just changing it into - I wanted to call it the Sonic Interocitor, but I'm not sure. And of course that pays homage to one of the best sci-fi movies of all time. The Interocitor was what Cal Meacham built when he received instructions mysteriously in the mail in the movie "This Island Earth."

Leo: Oh, wow. That is an obscure reference.

Steve: Well, I'll bet you we've got listeners who go, interocitor? I remember interocitor.

Leo: Oh, they love it. Interocitor. I know what interocitor - yeah.

Steve: Yup. So...

Leo: There's another guy in the chatroom says, is there anything that's guaranteed to work on cats?

Steve: How are cats bothering anybody?

Leo: I don't - cats are nice.

Steve: They don't bark.

Leo: No. They just - their pictures keep showing up on Google+. That's the only thing wrong with them.

Steve: Okay. So anyway, now, so listeners who want something which is not wimpy, which looks to me like it clearly works...

Leo: Or sets things on fire. They have a video on their site from a Tech TV show called "Invent This," in which they show their interest in dangerous stuff.

Steve: Yeah.

Leo: Wow.

Steve: So these are not wimpy solutions. Use at your own risk. They are available pre-built or also in kit form. You can buy the plans only and download them immediately online. Interesting site which I ran across because I was looking for a parabolic dish, and that led me to everything else. It's not the kind of thing I'm going to build. I'm going to build something very cool and very powerful. I have found 200w Class D, which is to say Class Digital switching audio amplifiers, one of which will be used for driving this thing. And as I pull the pieces together I'll end up sharing it all with our listeners. But in the meantime there are solutions that exist which do not look like they are wimpy.

Leo: No, they do not. No wimpy solutions here. Couple of people came up with the same thing for you. Mark Ping in Chico, California says you might be wrong about encrypting first name/last name: You said there's nothing preventing a service from encrypting our personal info. But imagine customer service searching the database for a caller's name having to decrypt the name columns before searching. That's not going to work. He says, everything besides name info could be encrypted.

And Simon Lyngshede in Denmark argues: In Security Now! 308 where you and Leo talk about a listener question, the listener asks why customer data isn't always encrypted on corporate servers. Your answer is it should be and that there's no overhead in doing so for the developers and the company. I'm currently working on a web shop for my employer, and I'm having a hard time seeing how we should encrypt all customer data without adding a huge overhead. If you store your customer data in an SQL database, aren't you forced to decrypt data every time you need to look for something? Let's assume we use your email address as the username. If this were encrypted, the database would no longer be able to do a lookup. You would have to decrypt every row on the accounts table to find the account you're looking for.

Similarly, you need to be able to do customer support, and customers are capable of forgetting pretty much everything, which means you need to be able to look through the database tables by name, address, phone, email, pretty much everything. Having the data encrypted would make it very hard to find anything. You could either have your application encrypt the input data and simply look for that, but that would mean that queries such as the "select" query, using your name "Steve" wouldn't work, not even if you encrypted "Steve" as the encrypted version would look anything like the encrypted version of "Steve." Could you share your ideas on how this would work? I think we probably get the message. Maybe it was a bad idea.

Steve: So my database behind my eCommerce system is encrypted. And we have, I don't know, hundreds of thousands of records. And Sue is able to pull up anyone's record instantaneously.

Leo: So does it decrypt the database and then do the search? Or does she keep a decrypted version in memory?

Steve: No, neither. The technology is the one you want. And I do not know what SQL does. But I do know that SQL does offer encryption. I just don't know at what level it operates. And I've never needed to know because I've never used it because I would never, ever use SQL for my own stuff. This database, my database encrypts it at the block level because encryption is much faster than reading and writing data. And so, for example, as the data comes off the disk, we just run - it uses Blowfish, as it happens, which is a very fast and secure cipher. It decrypts this block. And it may read, like, 16K blocks. So it decrypts it, and so everything the system sees, the system sees it as in the clear. Yet what's stored on the disk is absolutely pseudorandom noise. So if someone were to get into our system and somehow get that data file, it's of absolutely no use to them.

So it's certainly the case that, if you needed to, like, call an encryption function to decrypt a record, then that would be really bad, and it's not what I meant. And I agree with these guys that, if that's what they're thinking, then they're correct. You couldn't search the database. But assuming that SQL's encryption is implemented correctly - and again, I haven't looked at it, and I intended to by the time we had this question come up, but we changed our recording day...

Leo: Sorry about that.

Steve: That's all right.

Leo: I should mention we did that because - normally we do the show on Wednesday. We're doing it on Tuesday because we expect Apple to make some big announcements on Wednesday. So we flip-flopped you with MacBreak Weekly.

Steve: Yeah. So I imagine that next week or the week after we will have listeners who do know the answer to this question, so I'll punt to them. But it is absolutely possible for the file itself to be kept encrypted and for there to be zero performance overhead. And I'm hoping that's what SQL did. And if so, if we've got, for example, Simon, if you could look into that, see if there's a way to perform low-level encryption so that it's on the fly as the data's being written to the drive, then you should be able to perform lookups absolutely without any overhead or problem at all.

Leo: Well, Tom in Germany says: No, Steve, you can't encrypt everything. On Security Now! 308 you talked about the reasons why companies don't encrypt all their user data and mentioned there's no real reason why. I agree there is no computational overhead and that it can work securely for some services. These are services which provide infrastructure, but you are the only one with access to your data, for instance, LastPass. However, the vast majority of services don't work this way. You and the service need access to parts of your data. They therefore have to have the key, too; right? If a cracker gets the database, what prevents him from stealing the keys?

It's the same problem as with DRM on home entertainment products. As you yourself stated, it can't be secure because key and encrypted data are accessibly by the player, which can be accessed. If you exchange "player" with "server" and "accessed" with "broken into," you see there's no real difference between stealing plaintext DB files alone and stealing encrypted DB files and the keys. I'm a security advocate myself. I think the situation could be improved. But I respectfully disagree that it's possible to securely encrypt all data in all services. Please excuse my English, it's not my mother tongue. Thanks for the great podcast, and greetings to you and Leo, as well. Tom, your English is excellent.

Steve: I was going to say, Tom in Germany's English is much better than Steve in Irvine's German.

Leo: That's for sure true.

Steve: Okay. So, Tom, I completely agree with you. It's a function of what we mean by "secure." Is it more secure to encrypt the data before it's stored on the drive? And I'm sure you would agree with me that it is more secure. Is it absolutely unbreakably secure? No, because you're right, it is very much like the DRM problem, where we know we can't encrypt DVDs because the player has to decrypt it in order to show it to us, and that's in the hands of the consumer. For example, the attacks we've seen have been files stolen from systems. And those files, if encrypted, don't do the bad guys any good. If you could steal them post-encryption, then obviously the encryption is not helping you, and you're not getting more security.

Yet the cost is so low to store the files encrypted that my point was everything should be. There is no cost. It is not perfect security, admitted, because the system can decrypt the data itself. But it raises the bar substantially for the bad guys to get the keys because they could be stored on a different box, or they could be stored in a YubiKey offline encrypted container. I mean, there are ways to make the keys much harder to get to than the data. For example, my system, it derives the key on the fly, and it's nowhere available in the machine.

So, I mean, I may have over-engineered this, but I was designing this with security as a priority. So you can make it very difficult, if you want to. Perfect, no. As we know, you could argue there's no such thing as perfect security. But the cost is virtually zero to store everything encrypted if you can decrypt it on the fly. And I'd much rather have my files stored as pseudorandom data so that, for example, if someone came along and grabbed the hard drive, they've got nothing because the keys aren't there. And there are certainly ways to make that happen.

Leo: Our last question comes from Simon Hart in Burwell, North Cambridge, England. He's wondering about ISO files on USB keys: Steve and Leo, I've been a long-time listener to Security Now!, very much enjoy listening to it, blah blah blah. I was interested in hearing the episode.

Steve: Blah blah blah.

Leo: Well, everybody says nice things. Thank you. We appreciate your nice things. I was interested to hear the episode recently when one of your listeners recommended burning an ISO to a USB stick in order to create read-only media. I, too, have been playing with the possibility of using a USB stick for sysadmin. Actually I'm trying to get multiple ISOs on the stick so you could select the ISO using the GRUB loader from Linux. I, too, thought this would be a good way to create a read-only environment and was going to suggest it to the show. However, a doubt crept in before I did. Surely the reason to boot from read-only media is to avoid the cross-infection of suspect machines. But simply burning an ISO to a USB key does not make the USB stick read-only. It just makes the booted environment read-only. If I put the same stick into an infected machine, the infection can still place malicious files on the USB key, which potentially could cause another machine to be infected. Or am I missing something? Simon.

Steve: You are missing something, Simon, because that USB stick does not have a file system.

Leo: Ah.

Steve: That is, we're used to thinking of a USB stick as you just store files on it. So you're not storing the ISO file, that is, .ISO, in like a FAT or an EXT Linux partition file system or NTFS or anything. You're actually burning the ISO to the USB. So the USB stick does not have a file system that another computer could write to because it has an ISO CD-ROM file system, which doesn't have the ability to be written to. The operating system recognizes it as a read-only file system, and it's been created and enclosed, so it cannot be modified. So it's not a normal read-writable file system containing an ISO file. It is that file as the file system, so there's nowhere to write to.

Leo: Well, there you go.

Steve: It's a clever solution.

Leo: Clever.

Steve: Now, again, I wouldn't say that it is as secure as a hardware write protection because in theory you could have somebody modify the ISO file system. I mean, it's not going to happen. But in theory it could. So if you're a belt and suspenders person, I would do both: burn the ISO file system to the USB and have hardware write protection on that USB stick. And as we know, those are available.

Leo: Or just use a CD or DVD.

Steve: You could do that, too, yes.

Leo: Steve Gibson is the man in charge of the Gibson Research Corporation. That's why you'll find him at GRC.com. You'll also find that that's his Twitter handle, @SGgrc. And if you go to GRC.com/feedback, you can ask questions for our next feedback episode, two episodes hence. Are we going to go back next week to the Internet basics?

Steve: We're not going to immediately because something came up that just wound my clock.

Leo: Uh-oh.

Steve: Someone tweeted me, his name was Walid, or @Walid on Twitter. He pointed me to a security breach that has long been standing. I think 13 years is the number that stands out in my mind. This was a mistake that was found in the Openwall system. I think it uses Blowfish as a hash. The reason I bring it up is that it is a - I've so often talked about using generalities. I've talked about how hard it is to be perfect with security, how easy it is to make a mistake. It happens that this particular mistake, which exists in Linux systems all over the world, that has been in the code for 13 years, is a perfect example of something that people could look at day in and day out and never see what was wrong. It involves the C language. It involves some things that the compiler does for you that is really subtle. And I realized that it would make a perfect episode. I'm going to explain about types and character and integer types and this little bit of code and what it does. It's going to be a propeller-head episode, one of our ones that's going to make people think. But I can explain this so that everyone is going to be able to understand it and get it. And I'm just - I'm excited.

So the Internet series is going to be an ongoing thing. But when something like this or something really important newsy comes along, we'll interrupt that in order to deal with something else, with How the Internet Works series sort of happening to fill in the gaps in the background. So next week we're going to take a close look at, you know, we've talked about open source, the security of it. I've talked about how difficult it is for code to be perfect. And this is so self-contained and just the right size that I think our listeners are going to have a really fun time understanding a real-world example of how something that is just so obviously correct can be actually incorrect.

Leo: Great. I can't wait. That's next week.

Steve: Yup.

Leo: We do this show normally Wednesdays, 11:00 a.m. Pacific, 2:00 p.m., 1800 UTC. Wednesdays, live.twit.tv. Our next show will be from the new Brick House studios down the road.

Steve: Yay.

Leo: I don't know exactly what part of the studio we'll be using. And we'll be using Vidyo, by the way, Steve.

Steve: I'm ready.

Leo: Yeah. We'll get you ready on that. You should look fantastic in high-def. You can get this show in high-def starting next week, after the fact, at TWiT.tv/sn. Or, if you listen to the audio, we've got the audio, but Steve's got 16KB versions. He's the only source for that. So if you're bandwidth-impaired, or you've got caps or whatever, 16KB versions at GRC.com, along with transcripts, only place you can get those, and full show notes, too. GRC.com. And don't forget to get SpinRite while you're there, the world's best hard drive maintenance utility. Steve, we will see you next week on the all new Brick House edition of Security Now!.

Steve: Thanks, Leo.