SECURITY NOW!

Transcript of Episode #492

Listener Feedback #205

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

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-492.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-492-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We'll take a look at the security news. Fortunately, no big panics. But there is a great series of questions, 10 of them, from you. Steve will answer. We'll talk more about key escrow and more with Security Now!, next.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 492, recorded January 27, 2015: Your questions, Steve's answers, #205.

It's time for Security Now!, the show that protects you and your loved ones and your security and your privacy online with the guy who knows all, tells, all, he has a crystal ball, Mr. Steve Gibson of GRC.com.

Steve Gibson: Kissing my microphone. It's like right here in my face.

Leo: Why are you kissing it? Are you happy?

Steve: I'm happy, yeah.

Leo: Hi, Steve.

Steve: Hey, Leo. It's great to be with you again, as always in our 10th year of this weekly podcast.

Leo: Jiminy Christmas. Holy cow.

Steve: We finally do have a Q&A, since the world has left us alone. It's really been actually kind of quiet on the security front. There were a couple stories that just broke as I already had set up the podcast and made PDFs of notes and things. So, for example, there's a Ghost vulnerability that's just been - just was published in something called Openwall that looks like it may have remote exploitation capabilities. So we'll cover that next week. I didn't want to run around and go crazy.

Leo: Ghost is an open source firewall program?

Steve: Ghost actually is the name of the vulnerability. You know, we have to give them good, catchy names now, so it's, yeah, that's...

Leo: Right. Openwall. What's it called, Open...

Steve: Openwall.

Leo: Openwall, okay.

Steve: Yeah. Anyway, so we have a Q&A. And I'm assuming that things are going to stay quiet, and we will finally next week discuss the deanonymization of Tor and what that's all about. I've been calling that DeTor and promising it for quite a while, but we keep having holidays and new years and the Enigma Machine podcast that was very popular. So we'll catch up with a Q&A this week. A little bit of news. Not too much, actually, has happened, but some interesting stuff. And then we have great questions and some interesting points brought up by our listeners.

So I'm going to talk about a new marketplace that's open for Firefox; Google taking a bite out of Apple, also. And I heard you guys talking on MacBreak Weekly about some recent Apple updates that I also cover. And a note about Tim Cook and China. And I also heard you mention Vivaldi, which I had a note here about.

Leo: Yeah, yeah.

Steve: And I wanted to make a note to our listeners about the 10th Annual Podcast Awards. They're not getting by me this year. I thought, let's just swamp them. Let's show them what crowd power can do.

Leo: Yeah. I'm not a big fan of the Podcast Awards; but if you wish to win one, you go right ahead.

Steve: I like having them. Why not, yeah.

Leo: One year we won a significant number of them, and they removed us for political reasons. So I'm not - I just ignore them now.

Steve: No kidding.

Leo: Yeah. You can win and not get an award. So good luck. Good luck with that.

Steve: Wow, wow.

Leo: Actually, if everybody voted for Security Now! in every possible category, it'd be hard to ignore you; right?

Steve: It would be hard to ignore, yeah. I figure security - wait. I don't think there was security. There was, oh, science and technology and education. Although really we're not an - well, we're an educational podcast, but we're not a podcast about education. So I don't know if there are podcasts about education, if that's what they mean. But anyway...

Leo: Yeah. I'm boycotting this from now on.

Steve: Somebody tweeted that a new item had appeared under the Tools menu of Firefox. And I was at Firefox v35 when I first went to Help, and it said, oh, restart for an update. And that's like, oh, okay. So I got a one thousandth update, or one millionth or something. Anyway, I went up from 35 to 35.0.1, so hardly worth restarting. But sure enough, on the Tools menu is a new apps item which simply takes you to marketplace.firefox.com, which is Mozilla's marketplace for Firefox stuff. So it's a store that has lots of free stuff and also some for purchase, featuring apps designed for any device that runs either the Firefox OS or Firefox for Android or Firefox on any desktop. And they just say that it makes finding favorites or new apps easy. And, once purchased, you are able to use it across all of your Firefox devices and platforms. So, you know, sort of like in the iOS model.

So anyway, I just wanted to point people at that. I'm still, until Chrome solves the tabs problem, I am still over on Firefox. And actually the footprint problem, when I launch Chrome, it just gobbles memory compared to Firefox. So I'm still there, over on Firefox, and happy with 200 and some-odd tabs open. So, yeah, I sort of use it as my messy bedroom of stuff that I want to get to. I have tabs open from reference material that I was using when working on SpinRite 6.1, so that gives you a sense for, you know, SQRL took over, and we're now like at T plus 15 months of the SQRL project. So those are dusty tabs, but Firefox is holding them open and knowing that I'm going to get back to them soon as I get SQRL wrapped. And in fact I heard you talking about SQRL over the weekend. A caller to your Tech Guy show asked about it. So I just, you know, you and I will go back through this. I will soon be demonstrating it.

Leo: He wanted, as many have, to understand how it would look, how it would work from an end user's perspective. And I have to say I'm still a little bit unclear about it, too. I hope I didn't get it wrong.

Steve: No. The one thing that's neat is that you need no browser extensions or plugins. So you just run the SQRL client once on your desktop. And that registers it to receive any SQRL URLs. And then every browser you have automatically works. And you don't need a smartphone. You don't need a QR code. But you can just click on the QR code which the website presents in case you wanted to use a smartphone. But if you're on a desktop, you just click on the QR code, and you're logged in. Up pops a little dialogue from the client that is there to make sure you're you and not one of your kids or somebody. There's this new term that I hate, it's called the Evil Maid attack. In fact, the whole problem with Thunderstrike and Thunderbolt on the Apple is they've designated this an Evil Maid attack, meaning that it's an attack that an evil housecleaner could accomplish. It's like, oh, do we really have to call it the Evil Maid?

Leo: I think it's tongue-in-cheek, but yeah.

Steve: No, it's like it's the term.

Leo: So I did misstate it a little bit. So what I didn't understand is, so if you run the app on your computer, it registers - is it sqrl://?

Steve: Yeah, yeah.

Leo: It registers a protocol for SQRL. I think that's not commonly known is that there are many different protocols besides http://.

Steve: The famous one is mailto. You can have a mailto link, and that launches your email client with, like, oftentimes the subject and the to and from and so forth filled in.

Leo: So if you have this, if you download an app, you run it, I presume it generates kind of a unique userID for you; right? Is that what it does?

Steve: Yeah.

Leo: And it registers the protocol, the SQRL protocol. Do you have to keep the app on there? Or can you at that point delete the app?

Steve: No, the app is there to...

Leo: I think on most operating systems you have to because it has to run when you...

Steve: Exactly, yeah. And so it sits in the tray as a little icon. And so you create an identity the first time you run it. And it's sort of the thing where you have to see it to believe it. I mean, the demo has been running. I have it down at the moment while I'm rewriting a chunk of code. But users in our newsgroup have been playing with it for maybe about a week and a half. And one person wrote, he said, "You know, Steve, I knew what this was, I knew what it was going to be, but it takes my breath away to imagine that something this simple is also secure." Because that identity you create can potentially be the only identity you ever need for the rest of your life. And it just allows you to log in.

And of course you're right that the question is will this get adopted. There was a great blog posting that came to my attention yesterday, that'll go public on Thursday, where people are beginning to look at this and get it. And so my position was, with this concept, I just had to give it a chance. I have no horse in the race. I'm not making any money, as you did say yesterday to that caller. I just couldn't have this and not give it a chance. I mean, it blows FIDO away. The guy who designed FIDO said SQRL blows it away. So we'll see, you know. Anyway, it'll...

Leo: It requires, and this is what I said, it requires. unfortunately. companies with their own stake in all of this, unlike you, to adopt it. I mean, if Google and Yahoo! and Facebook don't adopt it, and they all have their own kind of, well, not Yahoo!, but Google and Facebook have their own authentication stuff they want to do, then it makes it hard for it to succeed. But I guess a ton of independent websites could adopt it. There's no reason...

Steve: I've already got queries coming from independent websites. We've got a group who have a full Drupal drop-in library so that any Drupal site can just add this side by side. The idea is that it doesn't replace username and password. It just sits next to it. And so the way I have felt this would happen is that there are so many sites where they would like you to create an account, but they're just not worth it to the user. How many times have we looked at, like, seen somebody's blog posting and gone to reply, and it wants you to create an account. It's like your email address and all of that, and you just think, no, I've got too many accounts. I don't want to have to come up with another password and username and blah blah blah. So that kind of site, or all of those kinds of sites, like WordPress, could simply make SQRL be an option. And you go, oh...

Leo: What happens if I've done that with that site, and I then I want to go to it on mobile? How does that get solved?

Steve: The identity is transportable. So, for example, on my client you can display your identity as a QR code and then snap it with either the Android or the iOS client. And that transfers your one identity onto your mobile device or devices so that same identity can then be used to log in there. And so, I mean, it is multiplatform and universal.

Leo: You'd have to have that SQRL client on your mobile device. Or no?

Steve: Correct.

Leo: What is the mechanism on mobile? Because I don't think you have - do you have that protocol mechanism on mobile?

Steve: Yeah, it's all there.

Leo: You do. So if you go to a site that you've previously registered a SQRL ID with, you go there, you see the QR code that says use SQRL, you tap that, and then it will launch the program, and it will go back to the SQRL, the thing. And that works on iOS?

Steve: So, yes. But there's two ways on a mobile. So if you're - the term I have is "same device login" or "cross device login." So say that you're at a hostel kiosk, or you're at a hotel executive lounge where they make some PCs available, and you want to log into your Southwest Airlines account. Well, the last thing you want to do is put your credentials into this alien PC that you have no control over.

So if Southwest adopted SQRL, then the Southwest login would simply have a little QR code next to it. So you would use your phone with the SQRL app to snap that QR code. And with doing nothing else, you are logged in on that PC without entering any credentials. The page changes, and it says, "Oh, hi, Steve. Hi, Leo. What do you want to do?" So your phone uses the QR code to identify the page that you're looking at on the computer where you are, and then it transacts with the website to assert and verify your identity. And then the website says, okay, we now know who is looking at that login page for Southwest, and you're logged in. So, I mean, anyway, you have to sort of see it to believe it. It's like, holy crap. And the crypto is solid. So anyway, you and I will do a demo. We'll go through this in a couple weeks.

Leo: That's cool.

Steve: You have a chance to use your brand new Dell XP whatever it is.

Leo: When it comes, February 5th. That's still a week off.

Steve: Okay, cool. So Google has - I did some more digging into Project Zero. We talked about it back in July when they announced it. And I remember we covered it on the podcast because when I was reading their announcement blog entry I thought, oh, yeah, this is all familiar. We did this on the podcast. But what's interesting is nowhere in there do they say we're starting a 90-day timer anytime we discover something. We notify...

Leo: Oh, that's interesting. Ah.

Steve: They do say - yeah, yeah. That came as a surprise. So they do say that we will only notify the vendor of the software in whose products we find a problem. And they assert that the goal is to increase the security of online stuff. And I did, in digging around, there was an example of a cross-site scripting vulnerability that Google found in their own stuff which they fixed in, I think, the number I have in my mind is like 17 days from the time it was found to the time it was fixed. So comfortably less than 90.

But the way this actually rolled out was that, as we have discussed because Microsoft has been caught by this a few times, Google starts a 90-day timer when they inform the vendor, and the exploit goes public 90 days later. So the way to think about this is that this is just sort of - this all fits the pattern we're seeing Google adopt. Google has taken the position that nobody is going to fix things without being forced to, thus the whole 2015 SHA-1 website certificate problem, where Chrome is going to start prematurely warning people that websites have certificates still using SHA-1 signatures, even though there's nothing wrong with that. It's like, okay, we're just going to push this so that it absolutely has been resolved by 2017, when Microsoft has decided they will no longer honor certificates that are so signed, so essentially two years ahead, and, you know, the others things that Google is doing.

And so this has sort of fixed that. It's like, yes, we're going to really tighten the thumbscrews. And the point I made last week was that this does - it's a little tighter for Microsoft because a fixed 90 days has to mesh with their 30-day window of opportunity, every-30-day patches. And if they miss it by a day, then they have a potential of exposure of 30 days because unless it's bad enough that Microsoft is forced then to do an out-of-cycle patch, they'd really rather have it fall on their calendar schedule.

And so in digging around in this I saw other people noting that this does create a dilemma for sort of the offline-style updating, where Google has this sort of on-the-fly Chrome is updating. I know that I've always got a process now running in my machine looking for updates to Chrome because it's just running in the background all the time. Well, that's not the way Windows has set up to handle updates. And arguably, the Chrome model is different than the desktop OS model. And you look at the Chrome OS model or the Google OS model, and here we have Android 4.3 and older, apparently not going to get updated, with really bad problems. So there's a little bit of a glass house sort of phenomenon here where Google is using the network to update the things that are easy to update on the network, but holding everybody else to a standard that you could argue they're not meeting.

So anyway, the good news is these three "zero-day," unquote, vulnerabilities that Apple had not patched by the time the 90-day clock expired, Apple was informed on October 20th, the 21st, and the 23rd about problems No. 130, 135, and 136. Google just numbers these things sequentially. And so 90 days later, on January 20th, 21st, and 23rd, they went public per this schedule. And so what we do know, and I missed the comment on MacBreak Weekly, apparently OS X v10.10.2 is, like, in beta, just getting ready to...

Leo: No, no, it's out. It just came out today.

Steve: Oh, it did?

Leo: Yeah.

Steve: Oh, okay, good. Yes, that's what I wanted to ask, whether that had happened.

Leo: Yeah, no, it came out, yeah.

Steve: Good. So what that does, it fixes, as far as we know, the Thunderstrike vulnerability, which unfortunately we're calling that the Evil Maid access vulnerability because the idea being that somebody with physical access to your machine could use the Thunderbolt interface, much as they could have always previously used the Firewire, we talked about how that is a - Firewire is a DMA-level access, and Thunderbolt gives that same kind of access, although there are security controls that Apple has probably now brought to bear or done some other mitigations. And we also believed that this fix would fix the Google Project Zero vulnerabilities 130, 135, and 136.

Although the point I was going to make was that, even though there was a window of exploitability, they weren't serious zero days. They were only exploitable if you already had code running on that machine. And if you have that, then it's like, well, okay, the horses are out of the barn. So it wasn't a big problem, and Apple has probably now closed it. And I also heard you note on MacBreak Weekly, that I'll say to our listeners, a new iOS update, 8.1.3. And I'm hoping that it does actually make iOS 8 more stable because it's a constant source of annoyance for me.

Leo: They're getting there. Inch by inch. Bit by bit.

Steve: Yeah.

Leo: By the way, it's not germane to this show, but I just was handed the Apple earnings reports. And Mac sales were up 14 percent, iPhone sales up 46 percent, iPad sales down 18 percent. So continuous tumble in the iPad. I know you're an iPad fan, but people are using their phones. With big phones, I don't know if they need tablets anymore.

Steve: Well, I certainly agree that the phone and the tablet has probably squeezed out the mini. I have two, and I would not buy them again. And the other thing is that I'm a big stylus person. And so I'm excited that Apple may be addressing that issue.

Leo: I agree, yeah.

Steve: Because none of them are any good. I have them all. And you know, when I say "all," I really do mean all. The active ones, the passive ones, the disks, the rubber nubbins, I mean, everything. And the problem is that it seems to be, from an engineering standpoint, unless you give it a strong enough signal, whatever that means, just generically, there's a long latency. Like it knows that it doesn't have a strong signal, and so it inputs a big smoothing filter which creates a smooth result, but with a lot of lag. And it's really unnerving.

Well, and in fact, a long lag smoothing filter means you can't go fast because it won't catch up with you. It's not like it's going to follow your path. It's always looking at a lot of history and averaging it. Which means, if you do something quickly, it just does a little small "nerch," which is the technical term. And you don't actually get the path that you followed. So you have to go really slow if you don't have a strong signal. If you do have a strong signal because you've got something big and capacitive on the screen, or something active stimulating the screen, then it's better. But none of it is any good yet. So I'm really, I mean, if we had - I always pronounce the company Wacom, but you guys are saying Wacom, as in whack-a-mole, so I guess maybe - I guess that's the...

Leo: I'll look. Somebody sent me a video where they say how to pronounce it. We know who you're talking about. Wacom, Wacom, yeah.

Steve: Yeah. And boy, does that - and in fact, I have one of those cute little early HP tablets, the TC1100, and that's Wacom or Wacom technology. There's another company whose tablet I have. Anyway, that technology, oh, is wonderful. I mean, it is full speed, smooth. And so I am assuming Apple will do it right when they do it and just, unfortunately, well, not necessarily unfortunately, put all of the other ones out of business, the early ones that they made their money, because I'd love to have - well, in fact, you've talked about how much you like your Note tablets.

Leo: Yeah, which has a Wacom digitizer in it, yeah.

Steve: Yeah, and it just works.

Leo: The thing that's missing on the Apple is - I didn't know about this lag thing. That's interesting. But the thing that's also missing is, and it probably has to do with palm rejection and other stuff. But the thing that's also missing is pressure sensitivity. So this will be interesting to see.

Steve: Well, there have been some interesting attempts. I have all the pressure sensitive styluses.

Leo: It's weird what they're doing.

Steve: Yeah. Some use sound. The earliest ones actually use ultrasonic sound, which the microphone could hear. And so, as you change the pressure on the stylus, you can't hear it, but the iPad's own microphone is picking up a signal ultrasonically, telling it how hard you're pressing. And then the newer ones use Bluetooth in order to communicate pressure through Bluetooth. And I imagine that's probably what Apple will do when they want pressure sensitivity. Like you, I don't care about that. I just, I mean, not only do I not need 1024 levels, two would be just fine, down and up. But I'm sure they're going to do it more for, you know, also serve...

Leo: Yeah, it's for artists, not for us, yeah.

Steve: Yes, exactly.

Leo: And I think, you know, they could actually put a digitizer in the screen. We'll see. I mean, they could put a pressure-sensitive digitizer in there.

Steve: So this is a little interesting. I'm not sure exactly what this means because some of the terminology is a little soft. But we do know that Apple's Tim Cook, the CEO, met with executives or representatives from China's so-called State Internet Information Office. And Apple has agreed to cooperate in China's audits of their iPhone, iPad - maybe just those two things, I don't know if it was the Mac - but their consumer-level products in order to assure themselves, that is, China, that there's no spying going on. Tim Cook reportedly told these executives that Apple products do not give data to third parties, saying, "We did not and will not provide a backdoor." And then the director of the department said, "The Chinese government needs to draw its own conclusions so that our consumers will be assured."

Leo: That's funny. There's a certain amount of irony in that Apple has put privacy in its products to make the Chinese happy. A little irony there.

Steve: Mm-hmm.

Leo: By the way, Apple's earnings, 75, almost, well, 74.5 billion, up 30 percent year over year.

Steve: Wow.

Leo: Big quarter. We knew that would be - 74.4 million iPhones sold.

Steve: The post-Christmas...

Leo: That's a lot of iPhones. Yeah, well, no, it is the Christmas, it's actually their Christmas quarter.

Steve: Right.

Leo: So it's their big quarter.

Steve: Wow. Yeah, and it was funny to hear you guys talk about the amount of iTunes Store app sales immediately following because...

Leo: Yeah, half a billion in the first week of January? Half a billion dollars' worth of sales? Wow. Wow.

Steve: Yeah, wow. And for people who used to love the Opera browser, one of the cofounders of Opera, lord knows what he paid for this website, or they: Vivaldi.com is their site, V-I-V-A-L-D-I dotcom. And actually Vivaldi's one of my favorite classical composers, very pleasant relaxing music. That's the name of the new browser. It will run on Windows, Mac, and Linux. It uses one of the nice products of Google's work. It uses the Chromium Blink rendering engine. So we're past the point where it's really feasible for someone to start from scratch and build a rendering engine. You just can't.

Leo: Unless you're as big as Microsoft. But any normal company, yeah.

Steve: Well, and do you think Microsoft rebuilt their rendering engine? Or do you think they just basically pulled...

Leo: You know, I've heard conflicting reports for this new Project Spartan, the new browser for Windows 10. At first Mary Jo and Paul said, no, it's the same rendering engine. But I just saw a report today that they've written a new one from scratch. So I don't know. I actually don't know.

Steve: And one also wonders how much from scratch it is, given that there's so much open source really good rendering engine technology. Remember that there was some question when Microsoft announced that they'd come out with a brand new internetworking stack for I think it was Windows 2000. It bore some strange resemblances to the FreeBSD stack, which was highly regarded at the time. It's like, okay.

Leo: Why write a new stack, really? Come on.

Steve: Really, why would you? Now, as Microsoft you really can't 'fess up to that. But so you scramble it up and change a lot of variable names and things and say, oh, yeah, yeah, this is - oh, in fact it had some similar bugs.

Leo: Oh. See, that's the giveaway.

Steve: Yeah.

Leo: When the bugs are the same.

Steve: Yeah.

Leo: So have you tried Vivaldi? What do you think?

Steve: I haven't. I like what they said. And the way they characterized it was, you know, we're not creating all new guts because why would you? Guts are done. What we care about is the user's experience. And they're deliberately aiming this upmarket to our kind of listeners, to power users. They have an interesting tab-combining feature that allows for economizing on tab real estate consumption. When you're, like, doing research within a given area, there's a way to combine it. Apparently it is heavily keyboard-oriented, so lots of shortcuts, so power users can use the keyboard to jump around within this browser and get a lot of fast work done. And there was something that I thought, okay, well, this is not exactly - this reminds me of TV screens I'm seeing now where it figures out the overall coloration on the screen and then backlights the TV screen sort of to give your living room that color. Well, this unfortunately sort of fades the chrome to the...

Leo: Oh.

Steve: Yeah, I know. It fades the coloration of the UI to fit the page that you're viewing. It's like, well, in that case it's going to be white whenever you're at GRC.com because all of my pages...

Leo: No, looks like it's still black, I'm glad to see. I'm on - this is Vivaldi rendering. You know, you've got to always test your page to see. Looks all right. Looks pretty good. Let's see.

Steve: Yeah, the menu works.

Leo: Your stuff's all there. Well, of course it is. Steve wouldn't make a bad site.

Steve: No, I've got simple HTML. I've got HTML from, in fact, I have embarrassing HTML from the early days. Stuff's there, and it works, even though I could go back and add - all my new pages use CSS extensively. But the old ones are still there, and they still render, thankfully. So anyway, for people who loved Opera, Vivaldi.com. It's at technical preview level now. Things like sync and mail and extensions are coming soon. So it's still - it's not there. It's not finished. But if somebody wants another browser - again, I'm not sure why. But for the features that it offers, it might be a thing for you.

Leo: It's pretty clean. It's pretty clean.

Steve: So I will just say, because it's here in my notes, PodcastAwards.com. Voting is not open yet. But it's only, right now, it's nominations. And I'm sure enough people have put Security Now! in nominations that we will be there. But for what it's worth, for people who want to make me happy, PodcastAwards.com can nominate Security Now!, I think for Science, Technology, and Education categories. And when voting opens, I will let you know again because it would be fun just to sort of show some feedback from our listeners.

Leo: And then all you have to do for the podcast URL is either use GRC.com or TWiT.tv. Either one would be fine.

Steve: Oh, in fact, it's funny. I saw a tweet. Someone, all they tweeted was "GRC.com or TWiT.tv."

Leo: Oh, they must be talking about this, yeah.

Steve: And I thought, yes, until you mentioned that, I didn't know what they were tweeting. It's like, uh, what's the question? So, yeah. That's exactly what...

Leo: You might - let's see. Please type something in the name text field. Please type something in the email. Oh, I see, I have to give them my name and address. Okay, well, there you go.

Steve: I think it ought to be TWiT.tv. I mean, that's, you know, it's your...

[Crosstalk]

Leo: Well, it depends what they want, is if they want a feed address, or if they want just a website address.

Steve: Oh, they know. I mean, if you put Security Now!, there's no one who doesn't know what that is anymore. I used to look at how many hits I got on Google, back nine years ago. We sort of...

Leo: I think just, yes, just do it. PodcastAwards.com.

Steve: Yeah. So I just wanted to close the loop and note that people who watched "Automata" that I referred to a couple weeks ago really liked it. That was the funky, I keep thinking Claude Van Damme, but it's not, it's the other guy. I can't remember. Anyway, interesting robots and sci-fi that you'll remember I said sort of was haunting. It had me thinking about it for days afterwards. Bunch of people did watch it and loved it. So I have another suggestion. What's happened is that, with all this Super Bowl stuff, my normal Sunday program has been wiped out for the last few weeks, so I've been poking around my Amazon Fire TV thing, looking for something to watch. And so I've been seeing things, I've been watching things I normally wouldn't.

Another movie that just came out, starring Ethan Hawke, is "Predestination." It got a 7.5 out of 10 on IMDB, which is pretty respectable. Rotten Tomatoes gave it an 80, which is surprising. And Metacritic gave it a 68. But a little over an hour and a half long, and it's based on a short story, as are many good sci-fi movies, by Robert A. Heinlein, of course a famous classic sci-fi author.

And it's funny because I didn't know that he'd written it until after, until it was scrolling through the credits afterwards. And I sort of thought, okay. Because, I mean, if you've read much of Robert A. Heinlein's stuff, it's absolutely written by him. It is an extremely complicated time travel story where, when you're done, you're like, whoa. And then the fact that it was contrived, you kind of can forgive it, because that's the way Robert's stuff tends to be. But it was a good piece, I mean, a very convoluted, paradox-filled time travel story, exactly the kind that he would write. And it's a movie called "Predestination." So I liked it. And I think our listeners will, too. So a little heads-up and thumbs-up on the sci-fi front.

And just yesterday, on January 26, a Bruce Young, whose title is Department Adjutant at Oregon's Department for Disabled American Veterans, sent email to Sue, because he didn't have another address, with the subject "SpinRite Is a Great Product." And he wrote: "I'm a computer professional, and I've owned the current version of SpinRite since v2.0 for my home PCs." And he says, parens, "(yeah, I'm that old)." It's like, hey, Bruce, that's all right, I wrote it, so I'm...

Leo: You were there since 1.0.

Steve: I'm at least that old. And he says: "I've restored several dozen 'dead' [in quotes] systems with this invaluable tool, most recently a four-year-old HP laptop that would hardly boot. And anytime a CHKDSK of my PC supposedly 'fixes' [he has in quotes] a drive, I run SpinRite on it, and the problem goes away. Sure, it takes a while to run. But if I kick it off before bed, even the biggest troubled hard drive is finished by the time I get home from work. And lest anyone think this tool is for techies only, the interface is easy enough to use that any computer owner can run it. Any time you need a beta tester for a new version, look me up. I'm ready. Thanks." And, Bruce, thank you for shooting that to Sue. She sent it to me, and I get to share it with our listeners.

Leo: Isn't that nice. We have no backup for you, Steve Gibson. I don't know. I just realized, we ought to clone you somewhere on the Carbonite cloud in case we lose you.

Steve: Yeah, I would like to be cloned just so I can get more done.

Leo: We need two of you.

Steve: Yeah. But the problem is then we would argue because we would both want to be doing everything. So I don't actually think that would work because my biggest problem when I had more employees and, like, a development group was that they were getting to have the fun of writing the code, and I was sort of the puppet master. And I wanted to be the puppet.

Leo: Wait a minute. You want to be the puppet and not the puppet master?

Steve: Yeah. I don't want to be a puppet master. I want to do the work. But cut the strings, and I'll...

Leo: Oh, I see. You don't want to do the management stuff. Oh, I don't blame you on that.

Steve: Right, right.

Leo: God, that is no fun. That is a thankless task.

Steve: I know.

Leo: But, see, you didn't do it right. The whole point for me of getting staff was so I would do the stuff I liked, and they would do the stuff that I didn't want to do.

Steve: And that is now my world. I have Greg for tech support, and I have Sue for the books. And it's like, thank you very much.

Leo: Yeah. So I hired editors, money people, HR people, that kind of stuff so I don't have to worry about that.

Steve: Yup.

Leo: Anyway, we've got questions; you've got answers. I installed the Mac update, and now I'm rebooting the machine. Let me pull the questions up here. Question 1 from Jay Mcgee in Washington, D.C. He wonders about this Elaine transcription service we use, Steve: Mr. Gibson, I have a need for a transcription specialist for a few videos. Could you forward me to your transcription specialist so I can send some business her way? Thank you.

Steve: So we talk about Elaine all the time.

Leo: Elaine Farris.

Steve: But we never - Elaine Farris. Now, her contact information has changed over the course of this. I just found her by googling, like, I don't know what, "audio transcription," and there were a few there, but hers was the only one that had an online form that I was able to sort of fill out and ask a question. And she got back to me, and then I found out, not only was she a transcriber, but she was THE transcriber. I mean, she, like, transcribes medical conferences where all the terminology is, like, she gets it correct. And she heavily uses Google to, like, research things. I know that because every so often I'll get email back from her while she's working on the transcription, saying, "Okay, you said this, and you and Leo were talking at the same time at this hour and minute into the podcast. What was that?" So she really cares.

Leo: That's nice.

Steve: And so it's fabulous. So she used to be On-SiteMedia.com, but now that's like "formerly known as." She now calls herself EDigitalTranscription.com, all one phrase, EDigitalTranscription.com. So that's where you can find Elaine. All that online stuff is still there. So you can send her email. You can click on a form and fill it out. She has a .mobi version, so EDigitalTranscription.mobi for her mobile version of the website. And as everyone who listens to this podcast knows, I could not over-recommend her. I mean, I can't overstate what it means to use her. She's not the cheapest solution there is, but she is the best, and that fits me. It suits my needs and the needs for this podcast.

Leo: I recommend Elaine.

Steve: Yeah, anything technical, or even if you just want them right. If you want them done right, EDigitalTranscription.com.

Leo: Very good. Now, by the way, other shows are transcribed on this network, and we use a different service, which I just asked. I just ran out in the studio, said, "Who knows who does our transcriptions?" And nobody knew.

Steve: Good. Elaine.

Leo: Yeah, so Elaine. But, no, this is a service that's been doing, not all of our shows, but I think about half of our shows now. Inspired by you and Elaine, we decided we ought to have transcriptions.

Steve: Well, it makes audio and video podcasts searchable. And that's so...

Leo: That's a big deal, yeah. That's the main reason we do it. Our next question from Elliot Kovacs, a 12-year-old listener in Wellesley, Mass. I love Security Now!, exclamation mark, exclamation mark. On a previous episode you said printers leave a watermark on paper. I was wondering if there was a way to see this watermark. I hope I get on Security Now!, exclamation mark, exclamation mark. I'll just call them "bangs" from now on. Bang, bang, bang, bang, bang, bang, bang. I'm only 12 years old, and I am very interested in security. Thank you. Elliot Kovacs. The end.

Steve: So it is the case...

Leo: They're invisible; right? They're not...

Steve: Well, they're not invisible. They're just very hard to see because they're yellow. So color laser printers use deliberately, on every page they put out, put yellow dots, like, on the paper. And the problem is you have to get a jeweler's loop and good illumination and just sit there looking for yellow dots. I'm not sure that a scanner, unless it was maybe a very high-resolution scanner specifically looking for isolated yellow dots, could find them. But the point is they're just not visible to the naked eye. The resolution of today's color laser printers is so high that the dot is very small. And yellow is just not that different from white. So you have to just look for them. But if you do, you will find them. And you'll think, oh, it's just some dust. No, that was put there on purpose. That is the serial number of the printer which printed that page. And I remember it sort of surprised everyone when this came to light. It was like, what? My printer is tattling on me? It's marking everything I print for, like, who knows why?

Leo: Well, I can tell you why. It's law enforcement.

Steve: Yeah.

Leo: So they've been able, for years you could always tell which type, you know, you could match a typewriter up.

Steve: Yeah, in the old...

Leo: They were so non-uniform.

Steve: Right, a mechanical typewriter, you're right. Every single one, very much like a fingerprint, would have slightly different characteristics.

Leo: So I'm sure when laser printing became popular, law enforcement went to the companies and said, hey, would you mind just putting a serial number on there, invisible, just in case?

Steve: Yeah. What can you do for us?

Leo: I don't think there's any harm to that. If you google "laser printer watermarks," you'll see close-up pictures of the yellow dots. But I suspect law enforcement knows exactly where to look and how to find that; right?

Steve: Yeah, I'm thinking.

Leo: It's probably in the same place on the page and all that.

Steve: Exactly.

Leo: And our transcription service, oh, now I forgot, I think it was Rich Hall at Perfect Security does the other podcasts. I don't know what the story is with that, but there you go. He's our transcriptionist for the other shows.

Question 3, an anonymous listener wonders: How do I get started in crypto programming? Well, first part is right. Anonymous, good start. I'm interested in crypto programming. I just don't know where to start since there are tons of tutorials on the web. I don't know where I should focus, which algorithm. I'm asking if you could point me in the right direction to somewhere I can learn how to program crypto properly. Thank you.

Steve: So the guru is Bruce Schneier. And I say that because he not only understands it, but he's a good teacher, as well. And behind me, let's see, no [muttering], right there. There. That is a copy of THE book. That's the one you want. It's Bruce's first book, "Applied Cryptography."

Leo: Wow.

Steve: And I pulled it off the shelf, and literally - somehow it acquired dust even though there's a shelf right above it - dusted it off, and just sort of flipped through it again to make sure that's the one I wanted to recommend. And, oh, my goodness, it is so good. So "Applied Cryptography" by Bruce Schneier, that's the one. Now, he's done several since. I think he did an "Applied Cryptography," and he did something about "Secrets and Lies" or something. He's done some others which are sort of more about the environment in which crypto is used. And "Applied Cryptography" is a little more of a cookbook.

But this is the one. You want "Applied Cryptography" because it is soup to nuts. It is across the board. It shows diagrams of all the algorithms, talks about hashes, talks about ciphers, symmetric and public key, I mean, it is sort of the introductory bible for somebody who wants a good foundation in crypto. So I recommend it without reservation. It's not inexpensive. And so I'm sure there are tutorials on the web that are free. Maybe they're worth what you pay for them. In this case, it absolutely is. That's the book.

Leo: And of course Bruce is still very active, talking about security and crypto. He's got a great website, Schneier.com, S-C-H-N-E-I-E-R, I think, dotcom. And we refer to him, Schneier on Security, we refer to him all the time. He's just a great guy.

Steve: Yup.

Leo: He's been on TWiT.

Steve: He does a great blog. And I'm seeing him catching a lot of media now. I'm seeing him being interviewed.

Leo: Oh, he's a god, yeah.

Steve: Yeah.

Leo: I just think so highly of him. And if you click his books link, you'll see all his books, and you can buy the books directly through him, get him a little extra money. But "Practical Cryptography" is there. Still, it came out in 2003, but still on...

Steve: Wait, but that's different than "Applied."

Leo: Applied, not practical, applied.

Steve: There it is.

Leo: There's "Applied."

Steve: It's that dark red one, yeah.

Leo: 1996, yup, available on Amazon in hardcover and paperback. Get it today.

Steve: It is just really, really good. It's a great book. It's the one.

Leo: It's in C, so you actually get code there; right?

Steve: There's a lot, yes, there's a lot of code toward the back of the book. But mostly, again, it is completely readable and understandable. It's what you want, if you want to just learn. And it's funny because we've talked about how slowly this stuff moves. Even though that book itself was written some time ago - the edition I have is the second edition. I don't know if there are more recent editions. But everything is still germane. I mean, because this stuff moves pretty slowly. So that's the one, "Applied Cryptography."

Leo: I was looking for my copy. I have it here somewhere. Not that I would understand a word of it. 1996 is the second edition. Still good.

Steve: It really is.

Leo: Yeah. Question 4 from Michael Zimmerman in Sydney. He's not alone in asking this, about extra keys and all of that: Hi, Steve and Leo. Just finished watching Security Now! 491 with your explanation of how, if mandated, a government backdoor could be provided using additional public keys to encrypt the random key used to encrypt the payload. This all makes sense, and I think we've seen the same idea used to provide the ability to change master passwords or provide temporary access such as application password access. My concern, among several, is this is impractical to implement. Let me put this up on the screen because it's such a long question, and I'll let you all, at least those of you watching on video, read along with me.

Steve: Read along.

Leo: Yeah. You've described the problem using iMessage as an example, but any legal crypto software could be inserted here. I want to send my message to Steve and Leo, and maybe the message is "Hi," or maybe it's my copy of "War and Peace." The message is encrypted with a random key, and a header is - well, not exactly a random key, but anyway, we'll - and a header is attached with Steve's encrypted version of the key. We also attach a header with Leo's encrypted version of the key.

Now, the government insists on access, so there is another header attached. The problem is, who is "the government," and how will the software know? I'm in Australia; Steve is visiting the U.K. Prime Minister Mr. Cameron; Leo's on a cruise - this is very accurate - currently passing Korea and China. According to Google, there are 196 countries in the world. Maybe I should have asked how many governments. We know that every one of them will want to be in on the action. The list of additional headers will spiral out of control, just like the list of CAs, any one of which can be used to decrypt the payload. Which one of them will be the weak link? Maybe the "Whatsnamia" official with a gambling debt, open to offers from anyone.

And if there were only one additional header key for the central controlling government body, who would administer that body, and who would deal with all the countries requesting to decrypt the header? Sure, my house is safe, and I have a front door key. But now my house has 196 separate front doors. All the best. Michael. He's describing one possible implementation, not necessarily the one that would be involved.

Steve: Correct. And in fact I will wrap up this with Question 10, where I'm going to submit what I imagine would be the actual implementation. But I included this because many people were a little thrown off by this last week. So I want just to make it clear that my intention wasn't to suggest that this is the way it would be done, but that doing something wasn't weakening the crypto. That was the entire focus, was to say that there are, I guess, there are two meanings of the term "backdoor." It can mean that the cipher is broken, that is, doesn't have its integrity in a way that most people don't know. And so that its lack of integrity, very much like a broken random number generator that is subtly broken, but no one can detect it. But the people who know about the way it's not quite as random as we want, they're able to leverage that fact. That's a perfect example of a defective cipher backdoor.

And so I wanted to make that distinct from the concept of the way multiple keys can be and are being used to create essentially, as Michael does say, multiple front doors. It doesn't require weakening the crypto in any way to create a means for something like mandated decryptability of content. So anyway, so I wanted to clear that up because many people responded as Michael did. I just chose his from among a bunch. I wasn't saying that I expect that this would be the implementation, but just that there could be an implementation that actually didn't weaken the cipher. Although you could argue, and as we brought this up last week, when you tell somebody a secret, it's not a secret anymore, even one person.

Leo: Right. I mean, you don't have to look too far to find the actual NSA technology which is in use today, called Skipjack, which is a key escrow system designed to do exactly what they want.

Steve: Right.

Leo: We'll, I think, talk about that later. Going to Question 5, John T. in Queensland's Sunny Coast doesn't think there are trivial true positives: See, in 489 a sysadmin explained they allow malware to enter their networks and then hope that host-based AV will clean up the damage. Well, that guy's crazy. Having worked in IT/infosec for over 20 years, in both management and admin roles, my first policy rule on any corporate proxy is to block executables. Period. No executables, ever.

That, in conjunction with a proxy-based AV scanning will block the majority of known malware at the gateway, instead of allowing it to be picked up and addressed by the host. If the AV signatures on a host can detect the malware, then it is likely that a proxy would also be able to do so, as they tend to be more powerful, have multiple AV rule scanning engines, and be more current than the hosts. Indeed, this process has become more difficult over the past few years as sites move to SSL, where an enforced man in the middle is required by the proxy to break open SSL traffic and scan it for executable malware content. But it's the only way for corporate networks to remain safe in this century. Regards, John, a long, long-time listener.

Steve: So I just wanted to present that counterpoint to the notion that we talked about before of local hosts cleaning things up. Maybe that's an approach, not the one that John suggests. And this sort of was brought to the fore for me because about a week ago someone wrote, a SpinRite user wrote to Greg saying that he had, I think it was an original PC XT, or he had a really old system, and neither SpinRite 6 nor SpinRite 5 would run on it. Could he get a copy of 4? And what has happened is, over the years, some really tasty op codes that I've wanted, like byte swap allows you to swap the bytes in a single instruction. And the original chip in the PC didn't have it. And but when the world had moved to 386s, I waited a while, you know me, I mean, I, like, waited a decade. And then I said, okay, it's safe for me to use a byte swap instruction on the PC. But not if you actually do have an 8088 or a 286.

Anyway, the point is that he was a SpinRite owner. I sent him a copy of SpinRite 3.1, which I was absolutely sure would still run on an original 4.77 MHz 8088. I couldn't get it to him. I kept trying. I zipped it, I encrypted it, I did everything I could. And whatever he had that was - the email just bounced. I could not mail it. So finally I gave it a quiet place to live on my server, and I sent him a link, and I said, "Okay, I give up. I can't get this to you. Here, you go get it yourself." And then as soon as he told me that he'd received it, I removed it. But, boy, it is not easy to get anything executable into someone's network these days.

Leo: As it should be.

Steve: Correct. It wasn't malicious, quite the reverse. But still he was well protected. And that's the way it has to be.

Leo: So if you had zipped it or somehow obfuscated it...

Steve: I did. This thing looked in the zip and saw that it...

Leo: It was smart. That's well done.

Steve: Yeah. You know, I even zipped it with a password. And it's like, because that's supposed to encrypt it so that you can't see. But maybe it looked at the table of contents? I don't know what it did. But anyway, it just still...

Leo: I know what you could have done because nobody would do - if you base64ed it, if you encoded it, it would just look like random text, then paste it into the message and say, un-base64 this, and you'll find an executable. That would have worked.

Nayef Alharbi in Saudi Arabia wonders about the safety of renewing a certificate: Steve and Leo, Security Now! listener since December 2012. Really enjoy learning and listening from you. I am wondering, is it safe to renew a certificate rather than requesting a new certificate? I know that making a new certificate is much more involved, but I feel it's more secure.

Steve: Now, that's interesting. It's something we, in all of the time we've been talking about certificates, this has never come up. So I love the question. What he's talking about is the issue of rekeying when you renew. And it is a choice. So we have a certificate. And remember that what a certificate is, is our public key, which is signed by a certificate authority, with an expiration date. So, and we know why expiration dates are important, because certificates can be bad, and we may have to blacklist them. But by having it expire, that self-blacklists, meaning that the blacklists don't have to be forever, they only have to be with some slack outside of the expiration date. Then we can prune off of the CRL, the Certificate Revocation List. Any management only has to live past the certificate's own self-demise. So that means that every few years you need it renewed.

Now, it isn't technically necessary for the owner of the site to generate a new key pair. It's certainly possible, and I've done this in certain circumstances, and I'll explain, to simply have the existing certificate, that is, the existing public key, re-signed and then redated. So you can do that. And so all that's really happening is that you're saying we're taking the existing public key and automatically the secret private key, and we're saying, as far as we know, they're still good. They have enough bits in them to be as secure as we need them to be for the expected lifetime of the certificate, which is never more than three years. So I don't feel like going through all the hassle of generating a new key pair. Let's just update the signature, re-sign it, and give me a few more years on the existing key pair.

Okay. The general feeling is, if you rekey and don't extend the expiration, that's always okay. And there are ways you might want to do it. For example, I recently rekeyed my certs and brought the expiration down to the end of 2015 because I want to stay with SHA-1 to help people get to GRC.com because you can only get to my website in any way now over TLS. So I rekeyed it short, and I re-signed, I had DigiCert re-sign my certificate with a tighter expiration using the existing key. I didn't rekey the cert. And they also gave me an SHA-256 cert on the original expiration schedule, which was I think in 2016 or '17. So the idea is you can say that's entirely reasonable because then the time horizon of your original public and private key pair hasn't been extended.

Now, it is the case that you could argue, hey, these public and private key pairs are strong enough that we're not retiring them because we think they're weak, we're retiring them just for the whole certificate hierarchy system. I think it's probably wiser, just as a matter of practice, it's only every couple years, to have your server generate a new public key pair so that, I mean, the danger would be that somebody could be working away in the background, trying to factor your public key. We know that's not really going to be a problem because it's secure enough. But the effect of rekeying is to take that public key out of service at the same time that you renew your certificate with a new expiration date, thus foiling any background long-term project of trying to crack your key, which again no one's really worrying about because we've got way more strength than we need.

So it's sort of a tossup. For example, say that you had a blogging site with low value to security. You're just going SSL or TLS because you wanted to, but there are not hugely valuable crown jewels being protected. Then, if it was inconvenient to rekey, you could just have the CA re-sign the existing key, and you're good to go for another two or three years. On high-value sites, I would say probably worth rekeying. But that is something we've never discussed. And it's interesting that people have a choice, especially if you've got a great CA, like I do with DigiCert.

Leo: Lee Whitfield at "32.924112, -96.7654598," which of course some of you may know as Dallas, Texas, offers some additional insight into CryptoWall 2: Steve, great show. Sadly, I sometimes just don't have time to get to the episodes for several weeks. This is one of those occasions. Episode 489 you discussed CryptoWall 2. I've had the chance to conduct forensic investigations on these systems and have something additional to share. Either way, CryptoWall is becoming more and more difficult to stop from propagating.

Yes, you can get CryptoWall from clicking on links in your email, from specially crafted PDFs and Office files. But there's an additional reason this iteration is so bad: You don't have to click on anything for this to run. The guys that made CryptoWall 2 were able to distribute it via ad networks, including Yahoo!'s own ad platform. If you have a web browser that allows the autoplaying of Flash content, and you happened to visit a perfectly good site that happened to contain a bad Flash ad, you could become infected. And on top of that, to avoid detection and discovery, the keys are issued via servers on the Tor network. This means it's difficult, or virtually impossible, to track these people down. Wow.

Steve: Yeah. So again, we know that there are two essential, well, mitigations. One is there's no substitute for using a sponsor of this podcast who you were talking about a few minutes ago, Leo, getting yourself backed up. Because if this thing gets into your system, all the files you care about are encrypted. And unless you have a current backup, you will be paying somebody to get your files decrypted.

The good news is they're pretty good about doing it because they want the reputation of, you pay them, you get your files back. So from the things that I've been seeing on the Internet, it's like they'll go to extremes. I've even seen situations where somebody, for some reason, is unable to provide them their money, or the money transfer didn't work, and they established a dialogue with these cretins and were able to get compassion from them, at least to the extent of arranging for them to accept the payment.

So the first is, you know, there's no substitute for having a backup of the content. And as we learned, I guess it was last week or the week before, a non-admin user can get recovered from Windows' own prior records of the changed files. So if you are running as a non-privileged, just a regular or standard user, then CryptoWall is unable, because it runs as you at the time of the infection, it is unable to reach into the privileged admin region of your system in order to also corrupt those incremental backups that Windows, the whole System Restore system that Windows is doing for people. So I would say, until we come up with some sort of a solution, you need to be backed up, and you really cannot run as a privileged user.

It's interesting, too. I'm seeing people who are having this problem at the moment with my Windows SQRL client because you have to be an admin in order to register that scheme, the sqrl://. One of the features I'll be adding, it's on my to-do list of cosmetic stuff, is if it notices that you're not admin, it'll itself prompt you for the admin, well, actually it asks Windows to prompt you for the admin credentials so that it can briefly upgrade you in order to get permission to register that scheme. I did the same thing with that freeware, Securable. Securable had to have admin privileges in order to install a driver, in order to access the chip's registers to do the stuff that Securable does. And I had it. It prompts you, it has Windows prompt you for the credentials in order to give it permission. So the good news is people are hitting this problem because they are running as a non-admin account. That's a problem you want to hit because you want to be running non-admin.

Leo: You could right-click on the SQRL program and say run as administrator.

Steve: Yes.

Leo: Just for that one time.

Steve: Yes, yeah. And if you don't, and then SQRL sees that it's unable to tweak the registry as it needs to, then it says, oh, give me a hand now, give me a hand.

Leo: Matt in London, with our Question 8, wonders how good Enigma really was: Steve, you threw some figures around about how many combinations there were on a three-wheel Enigma device, but what is that compared to a modern computer? Like when you compared SQRL's master key to the chance of the world ending every 11 seconds, how long would Enigma take to brute-force on, say, a Pentium 100 or a 2 GHz quad core? They built a mechanical machine to do it.

Steve: Well, yes. The bombe was a mechanical machine.

Leo: Far slower than any electronic computer would be.

Steve: Vastly slower. And many people wondered, how does Enigma's cryptography compare to contemporary crypto? And unfortunately, I mean, it was good for World War II. We would cut through it like butter. I mean, just not even a sneeze. So current crypto technology would just laugh this off, wouldn't even pause to crack the Enigma cipher. We've come so far. One of the reasons is that it just actually wasn't that complex. We did discuss how, for example, a state-of-the-art crypto like AES operates on this podcast, and I was able to explain it to people. But the nature of the way it works is vastly different from being a polyalphabetic cipher, which is all that Enigma was. I mean, we don't - you crack polyalphabetic ciphers in your first week of Crypto 101 in college.

Leo: Yeah. But it was good enough. And that's why they changed, by the way, that's why they changed it every 24 hours. They knew it was crackable with brute force even by hand within a certain amount of time.

Steve: Yup.

Leo: But it was good enough, and that's the point.

Steve: Oh, it was, I mean, given the computational, I mean, basically they had to create a special purpose computer. That's what Alan Turing did was a very, I mean, for the time, so sophisticated that nobody else understood how it worked. I mean, he sat down, and he designed a computer, an electromechanical computer, specifically to crack crypto, based on his understanding of the constraints that the cipher mechanism put on the encipherment. That is, it turns out, and this was the stroke of genius on his part, he recognizes, you know, the first simple thing that we talked about on the podcast was it immediately fell out of the design that no letter could encipher to itself. The machine made that impossible.

So that's, like, a simple constraint. But it turns out, when you look at it much more carefully, there's a huge, like, network of constraints. And if you build a machine to probe this network of constraints, what it would do is, and this is what it was doing when it was chugging away, it was testing potential rotor positions against a series of constraints and ruling them out. Nope, that won't work. Nope, that won't work. Nope, that won't work. And then when it would stop, it was because it found a series of settings that could function within the constraints that this bombe had been programmed for.

So then they would go, and they would use one of their Enigma machines and put that set of rotors in and see if the whole cipher made sense. The tiny piece they had would work, but the question was, does it all work? And so many of these were false positives. They would go back and push a button that essentially would say, okay, keep searching. And so it would start from that point and keep moving forward, looking for other possible rotor configurations that met the constraints that the bombe had been programmed in. The point is, our current crypto, we have technology that blasts it away. But back then, that was amazing.

Leo: That was good, man. Number 9 from Jonathan Blaine in Western Pennsylvania reminds us that the Astaro/Sophos UTM is free for home use: Steve and Leo, my wife and I have been listening to Security Now! for about two years. Really appreciate your efforts to educate on security. About two months ago, my boss and I were trying to figure out where my bandwidth was going. Actually, while I was trying to figure it out, my boss pointed me to the Sophos UTM, saying it was similar to the pretty amazing Palo Alto systems we use for work, but was free for home use.

I see from a search on GRC that Astaro was a sponsor back in 2006 through 2008, and heard Leo mention recently TWiT still runs the Astaro Firewall. It certainly isn't trivial to set up, and my family has had to bear with me as I got Netflix working properly on their various devices. But it truly is a powerful tool for the somewhat more technical group of listeners of Security Now!. It would be great if you had a chance to take a look at the free product and, if you think it's worthy, make a mention of it on Security Now! so adventurous listeners can experiment. Perhaps we could even share rules on common issues. Thanks again for everything. Jonathan Blaine, SpinRite owner, avid listener to Security Now!.

Steve: And I should say it is what I still recommend. I recommended it to a close friend of mine about a month ago.

Leo: I think you recommended it last week.

Steve: I may have...

Leo: On the show, yeah.

Steve: ...talked about the idea of taking an old PC with a couple of NICs.

Leo: Exactly, yeah.

Steve: And that's the one you want is the Astaro Security Gateway. I love the fact that he said it took a while to get Netflix working properly. And that's kind of what you want. I mean, you want tight admission of stuff into your network. And we know that NAT boxes, routers, do sort of a good enough job for most things. But Astaro really goes a lot further. It is updating itself with patterns, and it is checking to keep bad things from coming in, which a router has no function for doing. So I really like Jonathan reminding me to make sure that I tell people, you know, they were, what, I think they were the first sponsor for the podcast, weren't they, Leo? And they were with us for years.

Leo: They were our first TWiT sponsor, actually, as well as a Security Now! sponsor. Nice guys. Palen Schwab, who was the guy who bought the ad and has since gone on to other companies. And of course Sophos bought Astaro. But we always, in the ads, mentioned the free version of Astaro for download. That was part of the ad.

Steve: I'm glad to know it's still there. Stick it on an old PC with a couple of network interfaces, and look forward to having some fun pushing - it's got a lot of buttons and switches that you can flip.

Leo: And we do use Astaro UTMs throughout the place, and it's really been great. Makes me feel fairly secure. You know, we're safer than Sony.

Gert Eriksen in Denmark muses about master keys to cryptographic backdoors: Dear Steve, a very interesting topic, although I have some concerns about cryptographic backdoors. For me, the two biggest reasons for advocating for encryption is to ensure private communication, and to make mass surveillance unfeasible for private or government organizations. Nobody would argue that. That's right. Besides the obvious concern about a second front door, my concern is, if there is a master key to that door, it only requires one court order, and the master key to all communication in the past and future is loose. And this only for a single country's government. This is not a very pleasant thought, particularly for a non-American like me. Also, I don't know if that's true, but I'm sure Steve will address this.

In order to avoid this, I suggest that the second private-public key pair is generated individually for each user account, and the private key is stored by the service or app provider, Apple or Google. Then a court order will only reveal this account's communication and will not be a general master key to everything. I will very much like to hear your thoughts about this. I know there are some difficulty to generate, store and manage individual public-private key pairs, but the alternatives are much worse. Thanks for a brilliant podcast. Gert Eriksen, Copenhagen.

Steve: So I promised that the last question would address this. And this is the question. If this ever...

Leo: You've looked at Skipjack; right?

Steve: Well, yes. And if this ever happens, and we can hope it doesn't, but I'm skeptical, I think what would happen, the way this would actually get implemented would be sort of just turning the clock back legislatively, that is, legislation which required Apple and Google - and essentially we've already established that the cryptographic technology has already escaped. It is absolutely possible for two people to encipher so that nobody else can intercept their messages. We have that. But what would happen legislatively is that companies who were selling products that employed cryptography would be compelled to be able to respond to the FISA letters, compelling them to decrypt specific communications of their specific users or customers. And that they can do.

All Apple would have to do would be to maintain a master key and add that to every iMessage. Just as right now they provide keys to the recipients of iMessage, they would add their key so that they would be sort of a ghost recipient. And so this doesn't have to leave Apple. It doesn't have to be governmental. I mean, it's not - I didn't mean to imply any specific sort of structure when I was talking in broad generalities before. So I imagine, if something happens, that's what it would be is that companies, commercial entities selling products with cryptography, would be whatever they have to do. And that would be left up to them. But they would be, instead of now saying "We can't decrypt it, sorry, we don't have the keys anymore," legislation would say, "Oh, that's not good enough. If you're going to be selling crypto products, you need to be able to respond to specific orders to decrypt specific communications." And they could do that.

And so we as consumers would know that, rather than having things the way they have been for the last year, where Apple is like, okay, we can no longer crack your phone open, we're explicitly saying we can't do it, well, they used to be able to. Now they can't. They may be forced to do it again in the future.

Leo: Okay.

Steve: And that's, you know, I imagine that'll be the way it is.

Leo: I mean, you don't have to guess how it would be implemented because it already has been implemented by the NSA in this algorithm called Skipjack, which is now, since 1988, public, and you can look at it and see how they implement it. They use a key escrow strategy. And in fact Skipjack is in probably every TV manufactured today. So just so you know.

Steve: Yeah. The idea being that it'll be - the practical way for this to happen is that the entity that is selling a product that uses strong crypto will have to somehow, if such legislation happened, they would have to be able to respond, not that we cannot decrypt, but okay, here's the data that the court order required us to turn over. They would have to be able to comply with those orders. If legislation happens, that's probably the shape and the form.

Leo: Yeah. You could also require that whenever a key is generated by any of these products, that a second key is generated which is held in escrow by a third party, not Apple, not the NSA. Then that key can be turned over only on court order, that kind of thing. If you think about it, there's ways and ways to do this.

Steve: I don't think that'll happen. I think it'll be the way I said.

Leo: Well, I don't think any of this will happen.

Steve: That's all I'm saying. This is the way I think it'll happen.

Leo: Okay. All right.

Steve: It's basically just sort of turning the clock back a year. Sort of it's the way things were before.

Leo: Right. That'd be the easiest.

Steve: Yeah. So we'll have security, but...

Leo: It raises issues because then private companies have access to your stuff. So a better solution would be to have an escrow system that means that neither private company nor the government has access to it without an order.

Steve: Good point.

Leo: Yeah.

Steve: Good point.

Leo: But you know what, I think this is all pie in the sky. There's no political will to make this happen.

Steve: Let's hope. Please, please, please let's hope. Well, I don't know. The argument, you know, can we allow terrorists to have communications we cannot intercept. that's a tough one to - and then, of course, they yank out all the pedophiles and all that sort of - okay.

Leo: Steve, 10 in 10. Nice job. Once again, a hundred percent. Steve Gibson is at GRC.com. That's where you'll find his great program SpinRite, the world's best hard drive maintenance and recovery utility. You'll also find details on SQRL and all the other freebies he gives away all the time. He's always working on something new and interesting. Lots of information there. And of course 16Kb audio of this show, and transcriptions, as well: GRC.com.

If you have a question for our next Q&A, a couple of shows from now, you can go to GRC.com/feedback, or you can tweet him because Steve's also on Twitter: @SGgrc. And if you follow him there, he always puts up lots of great stuff. And anything with @SGgrc he seems to respond to. So that's another way to ask those questions. You will find full audio and video versions of this show on our website, TWiT.tv/sn, and of course wherever podcasts are aggregated. After 10 years, it's pretty much - I can't imagine a podcast client that doesn't have Security Now!, but you can always search for TWiT, and you'll find all of our shows. Thank you, Steve. Appreciate it.

Steve: Leo, a pleasure. Talk to you next week, and we'll do DeTor, how to deanonymize the Tor network, which was built specifically to provide anonymity. But it doesn't quite do it as well as we were hoping.

Leo: That's it for Security Now!.