
# Listener Feedback #107

**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-279.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-279-lq.mp3

---

**Leo Laporte:** It's time for Security Now!, the show that protects you online. And boy, I can't think of any better person to do that than Mr. Steve Gibson of GRC.com, the man who found the first spyware. Like a dinosaur hunter. Look, spyware! And not only did he find it, he coined the term "spyware," wrote the first antispyware program. He's also done all sorts of security goodness for all of us, including ShieldsUP!, his DNS Benchmark - Steve, it's good to see you again. I'm sorry I missed last week, but thanks to Tom Merritt for filling in. I think you had a good time.

**Steve Gibson:** Well, the only problem is, Leo, you missed a really fun and somewhat controversial episode.

**Leo:** What did you talk about?

**Steve:** We talked about implanting RFID chips in people; and, if I were to have one implanted, what would be my minimum requirements from a technology standpoint.

**Leo:** What a great subject. I'm sorry I missed it.

**Steve:** Really, really neat. It was really neat.

**Leo:** Thank goodness it's a podcast. I can go back and listen. We should note that,

while we will be here next week, that's the 22nd of December, we record on Wednesdays for a Thursday release, we will not be here the following week, the 29th, if you watch live or you download the podcast. But the good news is we're going to repeat the world-famous Portable Dog Killer episode. I can't wait. No, I'm sorry, Ozzy. My dog's in here. He's very upset. He said, "What are you talking about?"

**Steve:** Talk about his ears perking up normally. Oh, goodness.

**Leo:** You know he's a Papillon. As you saw, he has giant ears. And I don't know, I don't think it would kill him, but it might give him a headache.

**Steve:** He'd be jumping under the bed.

**Leo:** He said, "I'm getting out of here."

**Steve:** You wouldn't see him for a while.

**Leo:** I don't know what that sound is, but it's annoying. So, Steve, what are we doing today?

**Steve:** Today we've got a Q&A, our #107th Q&A. We've got, of course, some updates and some news. One really freaky bit of news that everyone has been tweeting me about, to make sure I knew about it - and you may have run across this, although it just happened - which is the claim that 10 years ago the developers of the OpenBSD security framework, specifically the IPSec stack in OpenBSD, 10 years ago these developers were paid by the FBI to build backdoors and deliberate side channel key leakage into it. So...

**Leo:** Oh, no. And, now, is this verified, or is this...

**Steve:** Well, we're going to talk about it. It's an interesting story, just broke, like, as we're recording this.

**Leo:** Holy cow.

**Steve:** Yeah.

**Leo:** We also have questions, 12 good questions from our listeners.

**Steve:** We got some feedback from last week. Not surprising, some feedback from the controversial embedded-chip-under-your-skin episode. And...

**Leo:** You know there's a side story to this RFID, and that's something called NFC, Near Field Communications. I'm sure you know about that. But I just got the new Google Nexus S phone, which has an NFC reader built into it. And it's kind of like, I think it's very similar to RFID. Of course it's very short range. You have to really get up right next to the thing. But you can pass the phone over a placard or a pay point or whatever, and data is transferred from that pay point into the phone. And apparently it's very popular in Japan.

**Steve:** Near field is interesting. It uses a different set of sort of parameters or terms from the original, I think it was the Schrdinger equations for energy transmission, where normal radio uses one set and has a certain characteristic in terms of distance versus power. Near field uses a different set of equations that essentially creates an extremely low-power, short-range connection which falls off very quickly. So it's different than just, like, low-power RF. It's deliberately designed to have, like, a different functional curve. And, yeah, I think we could see a lot of that in the future.

**Leo:** Very interesting. Okay, Steve. I see we have quite a few little updates here, including Microsoft's update.

**Steve:** You know, Leo, you started off saying that you were a fan. And it occurs to me that isn't "fan" short for "fanatic"?

**Leo:** Yes, it is.

**Steve:** Is that what it - I thought it probably was.

**Leo:** I am a fanatic.

**Steve:** Yes.

**Leo:** There are certain things I'm a fanatic about. And Ford has rapidly become one of those things because they've really, I tell you, this has been a banner year for the TWiT network and for all of our shows, in great part thanks to Ford and our other sponsors who've really helped us. You know we're going to build that - we're moving into that new facility, the 10,000 square foot facility, build all new studios. And all of that's because of our sponsors. So thank you. Tip of the hat. Microsoft had a December update while I was gone.

**Steve:** Oh, thank goodness, finally. This was the one we were waiting for. And the good news is this ends their updates for the year, this being December. They've broken their record. The total number of updates in any year was in this year, in 2010. This second Tuesday of the month, which we just had, addressed 40 vulnerabilities across 17 different security bulletins. They did fix the zero-day privilege escalation kernel vulnerability which we've been talking about and waiting for for a couple weeks. So the good news is that's done. That was the one that was frightening people because it had

the potential to be a means for malware to install itself as a rootkit, meaning that it would be able to get into your system in a way that, after it was in, no antimalware detection technology would have been able to see any longer. So the good news is that's fixed.

They fixed a bunch of vulnerabilities. Actually they completely caught up with all of the vulnerabilities that the Stuxnet worm had been using to get itself installed. Because remember that it was discovered that it was using some that were previously unknown. So that led people to believe that it was pretty sophisticated developers behind that worm. So absolutely, everyone should, as soon as they get a chance, make sure that their copy of Windows is up to date after this Tuesday, which was the latest occurring Tuesday in the month that we could have, since Wednesday was the 1st. Also just a small side note, and that is that Firefox has jumped itself forward to 3.6.13 and 3.5.16.

**Leo:** I don't know how they keep track of these numbers.

**Steve:** Fixing in the process 12 vulnerabilities, 10 of which were critical. So that was a good move there. Now, as I mentioned at the top of the show, the big sort of controversy - and I'm very much not a conspiracy follower. So I'm skeptical about all of this until it's been proven. Now, I was skeptical about Stuxnet and the very, very, what I felt were premature claims that this was targeted at Iran. It, of course, as we know, it turned out, once all the evidence was in, that it was almost certainly the case that that worm was deliberately targeted at Iran's nuclear enrichment centrifuge process control systems and was effective to some degree.

So the jury is very much out on the allegations regarding OpenBSD's Security Framework, or the OSF, as it's known, having been deliberately compromised. But here's what we know. Just yesterday, on the OpenBSD tech mailing list, Theo de Raadt posted the following.

He said, "I have received an email regarding the early development of the OpenBSD IPSec" - which is of course IP Security - "stack. It is alleged that some ex-developers (and the company they worked for) accepted U.S. government money to put backdoors into our network stack, in particular the IPSec stack, around 2000 to 2001. Since we had the first IPSec stack available for free, large parts of the code are now found in many other projects/products."

**Leo:** I have to say this seems suspect because it's open source. Wouldn't somebody notice this backdoor?

**Steve:** Well, see, this is the problem. I mean, open source, as we've discussed many times, you have a team who are working on some chunk of this. And first of all, I'm with you in being skeptical.

**Leo:** Yeah. I am not going to believe this story unless I hear some real confirmation of it.

**Steve:** Well, so the reason this has raised eyebrows is, he says, "Over 10 years the IPSec code has gone through many changes and fixes, so it is unclear what the true

impact of these allegations are. The mail came in privately from a person I have not talked to for nearly 10 years. I refuse to become part of such a conspiracy and will not be talking to Gregory Perry about this. Therefore I am making it public so that, (a), those who use the code can audit it for these problems" - potential problems I'll throw in - "(b), those that are angry at the story can take other actions; and, (c), if it is not true, those who are being accused can defend themselves. Of course I don't like it when my private email is forwarded. However, the 'little ethic' of a private email being forwarded is much smaller than the 'big ethic' of government paying companies to pay open source developers (a member of a community of friends) to insert privacy-invading holes in software."

**Leo:** Now, this is coming, this accusation came from Theo de Raadt, who's the founder of OpenBSD. So that does lend it some credibility; right?

**Steve:** Exactly. So he then quotes Gregory Perry's email. Gregory Perry is currently at GoVirtual.tv. And the subject was "OpenBSD Crypto Framework." And he says:

"Hello Theo. Long time no talk. If you will recall, a while back I was the CTO" - so the chief technology officer - "at NETSEC and arranged funding and donations for the OpenBSD Crypto Framework. At that same time I also did some consulting for the FBI, for their GSA Technical Support Center, which was a cryptologic reverse engineering project aimed at backdooring and implementing key escrow mechanisms for smart card and other hardware-based computing technologies.

"My NDA" - that's nondisclosure agreement. "My NDA with the FBI has recently expired, and I wanted to make you aware of the fact that the FBI implemented a number of backdoors and side channel key leaking mechanisms into the OCF" - which is the OpenBSD Crypto Framework - "for the express purpose of monitoring the site-to-site VPN encryption system implemented by EOUSA, the parent organization to the FBI. Jason Wright and several other developers were responsible for those backdoors, and you would be well advised to review any and all code commits by Wright, as well as the other developers he worked with originating from NETSEC."

**Leo:** I could guarantee there are developers, open source developers looking at all of those commits now. And we will know by the end of the day if there's any merit to this or not.

**Steve:** We'll know soon. He says, "This is also probably the reason why you lost your DARPA funding. They more than likely caught wind of the fact that those backdoors were present." I mean, here we're in, like, speculative land. So again, take all of this with a grain of salt, as I do this whole thing until, again, until we know more. But reading this, he says, "They more than likely caught wind of the fact that those backdoors were present and didn't want to create any derivative products based upon the same.

"This is also why several inside FBI folks have been recently advocating the use of OpenBSD for VPN and firewalling implementations in virtualized environments." Again, we have no reason to, I mean, this is all speculation. "For example, Scott Lowe is a well-respected author in virtualization circles who also happens to be on the FBI payroll, and who has also recently published several tutorials for the use of OpenBSD VMs in enterprise VMware vSphere deployments. Merry Christmas. Gregory Perry, Chief Executive Officer of GoVirtual Education."

**Leo:** Wow.

**Steve:** So that's what's known at this point. The good news is this is really on the radar and has come to the attention of the OpenBSD open source community. I'm sure that the people who were involved are being asked. We'll know more, I'm sure, for this podcast next week. This just happened. And I've been getting tweets and tons of email, and it's all over the place. So, you know, certainly...

**Leo:** No, ericduckman asks in our chatroom a legit question. Is this only OpenBSD, or does this code extend to any other projects?

**Steve:** Well, you'd have to carefully look at where any other IPSec and security framework code came from. What Theo was saying is these guys were first. This was 10 years ago. And as happens with open source, sort of by design, you can take chunks of it and put it in different places. And we see that all the time with OpenBSD, FreeBSD, NetBSD, you know, they've all taken overtly and, you know, saying thank you very much, they've taken big chunks of each other's work and incorporated it into their own operating systems. So again, it's going to take some time for the community to react to this.

The problem is that it - and we've spoken of this before. It can be extremely difficult to find these things, even when you're looking for them. That is, if they're talking about, for example, side channel leakage of key material, what that's saying is that, when the code was written, the developers, assuming that this were true, and again, I have no reason to believe it, it sounds very suspect to me.

But the way you would do this, if you were going to, is you would write the code in such a way, for example, that its timing was deliberately a function of the key which was in use at the time, so that, if you knew what had been done, then you would know how to look for subtle variations in the behavior, not overt like anything obvious, but subtle variations like timing or power consumption, but timing is the most easy to detect, where, if someone had said, okay, thanks for the check, here's what we did. And then you could, if you knew what to look for, you could use something like variations in timing to acquire information about the key that was in use on the IPSec crypto channel, that kind of thing.

So, I mean, really what you want to do is you want to scrap this and just write it clean. Otherwise you'll never really know for sure unless we can get statements from the people who say we have no idea why Gregory Perry is making these allegations. They are completely bogus. On the other hand, wouldn't the authors say that even if that was true?

**Leo:** Right. That's no proof. But that's why I love open source. You can verify this. You can look at it. You can tell.

**Steve:** Yeah, although, I mean, so many times I've drawn the - I've used the example of debugging. When my code is doing something that is unexpected, as the author of it, and even somebody else looking over my shoulder, we can, like, be looking at the code, and it looks fine because with code comes sort of an implicit assumption that what I've

written is what it's going to do. Which is why there's this strange phenomenon when you're stepping through the code in a debugger, you can come right up to the problem. And it's not until it actually happens in front of you that you go, oh. I mean, it's a left shift instead of a right shift. I can know that I mean right shift. But I wrote down left shift.

And so it's so easy to just sort of have your eye step past it with the assumption that it's correct. So, similarly, analyzing code to discover something that a programmer deliberately did and deliberately obscured, I mean, presumably, if this was really done, it was done in a very clever way. So I could fully believe that someone, if this was done, could have taken a block of this as a module and just lifted it in and moved it into other, repurposed it into other operating systems, where it was a functional module known to do the following. And it really could have spread. So anyway, we're at the beginning of a very interesting story that we'll certainly be following for our listeners as more develops.

**Leo:** I'm going to call BS on it, but we'll see. Obviously we have to check.

**Steve:** I'm with you. I mean, the whole thing. I mean, first of all, I don't imagine that the FBI would pay open source developers to do this because this is not the kind of thing you can keep secret.

**Leo:** I'm sure they'd pay commercial developers to do it, but that's why I recommend open source for stuff like this.

**Steve:** Well, or you could imagine that they would have somebody working for them on the inside.

**Leo:** Maybe, yeah, committing that way, yeah.

**Steve:** I mean, that's the way you would imagine this would happen if that was really going to be done. So, anyway, it is a big story. I wanted to address it for our listeners.

**Leo:** Oh, absolutely, yeah.

**Steve:** And we'll certainly follow it. Many people have written to ask about the denial of service attacks on all kinds of entities who have been involved in many different ways, whether they were DNS suppliers - well, involved with WikiLeaks. Whether they were DNS suppliers; donation carriers, you know, PayPal; whether they were hosting providers; I mean, pretty much anybody who has in any way been associated with WikiLeaks, who has attempted to distance themselves from WikiLeaks as a consequence of this huge controversy, has found themselves the victim of denial of service attacks of varying strength, durations, and so forth.

What's interesting is that a simple-to-use denial of service attack tool called LOIC, which is an acronym for Low Orbit Ion Cannon, was created. And LOIC operates in a very well-known and almost traditional fashion for a botnet. It hooks onto an IRC channel. It joins an IRC channel in an IRC chatroom. And then the bot herder - and that's called the

Hivemind, the LOIC Hivemind. Then somebody who is controlling that channel issues commands which will be received by all of the bots that are listening in this IRC chatroom and will then go attack whatever target has been given.

Now, at one point as many as 1,200 different bot clients were logged into the chatroom. The problem for them is that they're making TCP connections, which are not spoofable. There's no way to spoof those. Meaning that anybody under attack is seeing standard Internet TCP connections from these bots. So without getting at all into the politics and ethics of this, I want to stay out of that question, it is the case that anyone who is contributing, who feels they're contributing their bandwidth for the sake of the anonymous group, as it's called, that is attacking people around this whole issue. Their IPs are exposed. And several of them have already been pursued and, I mean, legally pursued.

So it's certainly the case that this is not a sophisticated DDoS client. Many people have been surprised that websites have been so easily brought down by a relative small number of clients. This is only in the low thousands and high hundreds. And so one thing worth mentioning is that denial-of-service attacks are not difficult to launch. I mean, they are pretty easy to launch, in fact. And sites are relatively easily brought down. I mean, we had MasterCard brought down, Visa brought down, Amazon was having trouble, smaller DNS sites that had been involved had been brought down.

PandaLabs has a really nice blog that has been following virtually blow by blow, they've been watching the management of these botnets associated with the WikiLeaks attacks. And I created a short URL because theirs unfortunately is really long: bit.ly, so it's a bit.ly URL, bit.ly/hyGLpy. That expands to the Panda Security, PandaLabs blog. I also checked with Google. And at the time, at this time, if you put in "PandaLabs DDoS WikiLeaks," and I think I put in "LOIC," you can also find it through Google. But if anyone's interested, it is fascinating. They have essentially a complete chronology, a detailed chronology of who's been attacked, how hard the attack was, how long they were held off the 'Net, when they got back on, and so forth. So if anyone's curious, PandaLabs has done a great job of tracking that from the beginning. Which is interesting.

**Leo:** Yeah, yeah, absolutely.

**Steve:** There's even a JavaScript version of this LOIC, the Low Orbit Ion Cannon?

**Leo:** That tells you something.

**Steve:** You could even run it on your iPhone, if you wanted to. And I'll mention also that, when I was visiting Mark Thompson a couple weeks ago at his place out in Phoenix, he's got, I think it's 15 megabits of upload on his cable modem. He's paying for a high-bandwidth cable modem connection. But, I mean, that's a ton of bandwidth.

**Leo:** You get a few hundred of those, boom.

**Steve:** Exactly. That adds up very fast. And that's much more bandwidth than most websites are used to dealing with, like on a saturated, focused all at once sort of basis. IE9, Microsoft's forthcoming Internet Explorer 9, which is currently in beta, I just wanted

to mention does have a do-not-track technology of some sort. It will be present in the browser. It'll be disabled by default. But it will be something that people can turn on. And I'm sure that we'll have something shortly in Mozilla's Firefox, as well. And apparently Chrome's security model is causing developers some bit of problem, just for implementing these kinds of things. We do have something called NotScript for Chrome, which I wanted to mention that I know of. I've not taken a look at it yet. But many of our listeners have said, hey, Steve, we know how much you are in love with NoScript for Firefox. Well, there is NotScript for Chrome. And the developer mentions that he had to jump through some hoops in order to implement this for Chrome because Chrome's own high-security barriers were fighting this kind of functionality. It wasn't something that was easy for him to shoehorn into Chrome, but he's managed to.

**Leo:** And of course this is going to become more relevant with the Chrome OS, which is entirely based on the browser.

**Steve:** Exactly. There was a note also that the SANS security letter had, saying just that the UAE, United Arab Emirates, authorities can now decrypt BlackBerry communications with a court order. And their little blurb said that the United Arab Emirates' Telecommunications Regulatory Authority now has the key for BlackBerry services. This means that the authorities can decrypt and monitor BlackBerry communications after obtaining a court order. BlackBerry's parent company, Research In Motion (RIM), has reached a similar agreement with authorities in India.

Well, there was a link in the SANS article to a website with a broken link, or a broken page, when I clicked on it to follow up because I didn't understand what this meant. This says that they have a key, but they require a court order. But if they have the key they wouldn't require a court order. Or it's not clear what it is they have to go through to use their key. I mean, I wanted to get something more rigorously technical than what was in the SANS newsletter. At the same time, I didn't want to ignore this note because this is interesting to me from a crypto technology standpoint. So as more is known about this, I will let our listeners know because we've been following this whole issue, which I find really interesting, about what it's taking to do this.

We know that, if software is installed in RIM's servers, then it's possible for RIM as a man in the middle to perform the decryption. But RIM has said that, so long as people are using the corporate servers, which have no dependence on RIM at all, that various entities, the government entities would have to go to the corporate endpoints and see about installing some sort of third-party technology there, or RIM's technology in those third-party servers. Anyway, this is just a huge mess. But as we know more, we'll certainly let people know. And I did want to note that, unfortunately, DoubleClick and Microsoft, that is, the advertising server, rad.msn.com, were both found to be serving malware advertisements recently.

**Leo:** Yeah, I saw that. Wow.

**Steve:** Yes. Malware got into the advertising stream. This was a banner ad which was essentially using heavily obfuscated JavaScript to exploit at least seven previously patched vulnerabilities in Adobe Reader, in Java, and in Internet Explorer, in order to install something called HDD Plus, which was sort of semi-ransomware. It would inform users that their system had serious errors which required the premium version of HDD Plus in order to fix their system. So that was found and fixed. And of course Google

jumped on it immediately, Google being the parent of DoubleClick, and are trying to come up with a way to prevent this from happening in the future, which of course would be a good thing to prevent.

**Leo:** Yes.

**Steve:** And quickly, a little bit of errata. My own Sue, who handles sales support for GRC, messed up her filters in Eudora Monday evening, and yesterday morning wrote to me and said, Steve, I don't seem to be getting any email. Well, it turns out that I went down to see what was going on, and she'd mis-implemented some spam filtering, which it turns out triggered a bug in Eudora which was deleting any email that she received in such a way that it didn't even go into the trash, where we could have recovered it. It just went into nowhere, into oblivion.

So I wanted any listeners to know, if anyone had sent us email for any reason between Monday night and around Tuesday at noon Pacific time, we never got it. Unfortunately, it's nowhere. The way we had her email configured, we've changed that now, but it was pulling it off the server forever and deleting it. So we didn't receive it. I'm sure, I mean, we feel awkward about this because we hate not responding to anyone who has sent email to us. But just it was lost. So...

**Leo:** These things happen.

**Steve:** It does. It never happened to us before, and now we've taken measures so it won't happen again. We're now keeping email on the server even for Sue's accounts. Mine I do have kept on the server. But Sue was just deleting hers. So we had no problem until no.

**Leo:** This is why I like IMAP. Of course, if you had had IMAP, you probably would have deleted the originals, so never mind.

**Steve:** Exactly, it could happen, too. Also, the blog for Adobe tracking the development of Flash, I mentioned before that Flash v10.2 was supposed to be using much less processing power. And I had mentioned that in the context of my own experiments with Flash on a battery-powered laptop, where I was surprised when Flash was jumping around on the screen, how much battery my laptop was burning. Its estimate of, oh, you've got eight hours left dropped down to, like, an hour and a half when I had Flash running in the browser.

The good news is we're beginning to see some metrics from this. And they're claiming to use as little as 10 percent of the previous CPU and system power under this new v10.2. So the good news is they're really focusing on power consumption as a consequence of wanting to get Flash onto handheld devices. And it looks like they're being very successful with that.

And I got a kick out of something that I ran across in the mailbag that I just wanted to share with everybody. A listener of ours, Mack Morris, says thanks again for the great podcast, and tell Leo that I think his Irish accent...

**Leo:** Is terrible.

**Steve:** ...is the best of all the ones I've heard so far.

**Leo:** Is this guy Irish? Because that's the first problem.

**Steve:** Maybe not.

**Leo:** Maybe not. I love it. We have an Adobe Gotcha! of the week coming up, too, in the questions, speaking of Adobe.

**Steve:** Yes. We do. I did want to share a fun SpinRite story. I mentioned a couple weeks ago when I didn't do a SpinRite story that I had run across a number of really fun ones. This one is "SpinRite Saves the Broadcast." Our listener Sean McStay says, "Hello, Steve and Leo. I work for a mobile television production company that specializes in sports broadcasts. On Wednesday, the day before Thanksgiving, I was working an NBA broadcast in Oklahoma City. About 12 hours into the day my technical director called me to the production area of the truck and asked me to listen to something strange.

"There is a touchscreen computer that is the user interface for the production switcher that switches the entire broadcast. This computer was making a strange, high-pitched sound, but otherwise seemed to be fine. The technical director and I both thought that a worn-out cooling fan was probably the source. Other than the annoying noise it was making, I really didn't think much about it and just made a mental note to get a replacement fan.

"Well, I bet you know where this is going. After returning from our meal break, the technical director let me know that the touchscreen computer had locked up. Thankfully, he had already loaded his entire show into the switcher mainframe and didn't really need the touchscreen computer for the rest of the day. Now, fearing that it might be something more ominous than a cooling fan, I took the touchscreen computer back to engineering to give it a closer look. It wasn't a cooling fan. The hard drive was the source of the noise, and I knew I was in trouble.

"We maintain a very expensive service contract on the switcher, and I called the manufacturer in California. Now, realize that it is late Wednesday afternoon, the day before Thanksgiving. I had a show in Pittsburgh on Friday, the day after Thanksgiving. I knew that getting a replacement in time was a very iffy proposition. The manufacturer of the switcher did have a menu panel in hand and was able to get it out that day, but could not promise that I would get it in time on Friday.

"Now, like all careful truck engineers, I do have other ways for technical directors to get their shows loaded. But it's not convenient for them at all. I also had a pretty recent image of that particular drive. But I thought that if the hard drive was going bad, that image might not do me any good." Meaning he wouldn't be able to load it on the hard drive.

"Friday morning in Pittsburgh I told the technical director that if the computer came up at all, he would need to get his show loaded quickly. I did not trust this machine to remain

working for very long. I had a momentary good feeling when the computer made it past the POST (Power On Self Test), then displayed the Windows splash screen. But my hopes were quickly dashed when the computer bluescreened a few seconds after displaying the Windows logo. I sent the technical director back to my engineering computer to load his show, and I started SpinRite working on the menu computer.

"About three hours and 20 minutes later, SpinRite had finished. It displayed two unrecoverable sectors, but did not report anything else was amiss. Even though the drive was not making the unpleasant sounds that I had heard in Oklahoma City, I did not have a good feeling about it. To my pleasant surprise, the menu computer booted up and loaded the switcher application without a problem. We went through the whole show without any issues.

"I'm still surprised that SpinRite was able to take a drive that sounded like a small metal lathe and make it work again, but it did. I have since replaced the menu PC with a new one. I wanted to acknowledge that your product had bailed me out when I knew that I could not count on getting a replacement in time. Add my name to the list of clients who have been saved by your very functional product. Best regards and happy holidays to you and Leo. Sean McStay, St. Louis, Missouri."

**Leo:** Thank you, Sean. I am a little surprised that it worked, actually, because that sounded like it was a hardware issue.

**Steve:** Sounds like bearings in the drive with that kind of a high-pitched sound. And that could create some vibration that would cause the heads to have a problem. So glad it got it fixed.

**Leo:** Yeah. But I think he was right to replace it.

**Steve:** Yeah.

**Leo:** Who knew how long it would last.

**Steve:** And often SpinRite is, like, you use it just to pull your butt out of the fire one last time, and then that's all you need.

**Leo:** Right. And now, ladies and gentlemen, 12 questions good and true for our friend Steve Gibson. Are you ready, my friend?

**Steve:** And comments from our listeners, too. So, yeah.

**Leo:** Comments, questions, thoughts, Adobe Gotcha! of the week, that kind of thing.

**Steve:** Good stuff.

**Leo:** Start with Robbee Nelson, Raleigh, North Carolina. He says he may have found an alternative to the PayPal virtual credit card, which we were bemoaning the loss of. PayPal's discontinued those one-time-only, one-use credit cards. He says it's called ShopShield. I may be your biggest fan, Steve, and I appreciate everything you do, especially the great SpinRite. I JUST LOVE YOU MAN! he says in caps.

To be more specific and to the point, on October 7, 2010, Episode 269, Listener Feedback #102 (I used the GRC search), you sent out "A plea or question to our listeners, who are spread far and wide. If anyone knows of a replacement" - of the PayPal plug-in, which was discontinued. Well, I ran across this site, shopshield.net. I did some online checking, and ShopShield is highly regarded by the Identity Theft Resource Center. He says it's a nonprofit nationally recognized for providing education and resources to prevent identity theft, idtheftcenter.org. They review it on their site. He says: I listen to each episode every week, but I don't remember any feedback about a PayPal plug-in replacement. Just wanted to see what you think of ShopShield and if any other listeners have similar findings. Keep up the great work and remember, I LOVE YOU, MAN! Cute, that's cute.

**Steve:** So I wanted to share this immediately with our listeners. There is nothing I want more than a one-time credit card solution. I have not yet - I just ran across this this morning as I was pulling these together for the Q&A. So I have had no chance to do any research into the site. It looks legitimate. I mean, on the surface it looks like a good thing. And everything they're saying sounds right. I don't know quite how they accomplish what they say they're accomplishing because they talk about protecting your name and your mailing address and shipping address and, like, all personal information. And that's like, well, okay, how can you do that? Because, for example, Google's, I'm blanking on the name, Google's...

**Leo:** Google Checkout.

**Steve:** Checkout. Which I use and like because they're, in sites that offer that, it's one click. And Google does perform the transaction with someone who accepts Google Checkout. What we need is a system where someone who doesn't, like, accept ShopShield can still be used. So...

**Leo:** I presume that they use credit card numbers. I don't know, I'm looking at it right now.

**Steve:** Yeah. I have not pursued it. You'd need to - there's a free trial. And believe me, by this time next week I will have a tune-up on ShopShield. I did want to share it because many...

**Leo:** It seems like a good idea.

**Steve:** It really does. Many people have asked for it. So I've got my fingers crossed, and I'll give everyone a tune-up next week. In the meantime, anyone else who's interested is welcome to do their own pursuit of this, as well.

**Leo:** Yeah, we'd love to hear what you think, get your feedback. Question 2, an anonymous listener with a great question: Steve, thank you for your podcast. I've probably listened to all of them. I find them informative over the years. I've been involved in discussions with my work colleagues about which encryption algorithm to use on a low-powered CPU. It's running roughly 1 MIP, which is not very much in modern terms. One of my colleagues suggested RC4. It's simple to implement and won't take up too many CPU cycles. The device will be battery powered, so we need to keep the number of instructions to a minimum. What are your thoughts on RC4 for that?

**Steve:** Well, this is sort of interesting relative to the discussion we had last week of RFID crypto. Essentially, to tie this into that, what I was saying was that an RFID chip which simply blasted out a fixed ID every time it got pinged certainly fell far short of my own requirements for the crypto that I would allow to have implanted in me because we really needed - we needed real crypto. We needed something where no snooper could just listen to an RFID chip emitting its ID and then clone that and emit the same thing. So that requires some kind of useful crypto. And in something which is being powered by the radiation being emitted by a reader, you need to have something that's also very low power, very low computational overhead.

RC4 is even now, to this day, a really good cipher. It got a bad rap from its use in the very first implementation of WiFi crypto, the so-called WEP, Wired Equivalent Privacy protocol. But it wasn't RC's fault that WEP was so badly broken. RC4 is very simple. It's well known. It's a trivial algorithm. And when it's used properly, meaning that you discard the first chunk of pseudorandom data that it provides, and you're very careful about the way you seed the algorithm every time, if those criteria are met, the stream that it produces is extremely robust. It produces a pseudorandom stream of data which you then XOR with your plaintext to create the ciphertext.

So I really do think that, again, as long as somebody who really understands crypto is the implementer of this, I think it makes a lot of sense for a low-power use, strong crypto in a system where you have either low CPU power, low battery power, low speed, whatever it is. There's nothing wrong with RC4 as long as it's used correctly.

**Leo:** You give it RC'll of approval.

**Steve:** I do.

**Leo:** Yes, you do. Lance Reichert, itinerant engineer in upstate New York, says, I need to convince customer service that email is public. Recently, one of my credit card companies had the idea to show me how convenient paperless statements would be by giving me a temporary enrollment. One of the "features" of these paperless statements was a monthly email announcing the availability of my online statement and detailing my outstanding balance, minimum due, and due date. They were agreeable enough to remove me from the program immediately upon request, but were unwilling to accept that the practice of putting customers' balances and due dates in emails breached those customers' financial privacy and ran afoul of the Consumer Data Protection Act. They seemed to think that, since I had to log into my email server to collect my email, it was as secure as my email password. Is there

any compelling argument to offer to them that between their server and mine, email is publicly available to anyone who cares to read it? Signed, Lance.

**Steve:** You know, I've had email like this from our listeners before. I mean, we have of course created an educated listenership of people who really get this stuff. And they find themselves frustrated when they're trying to explain to people who are offering insecure services the nature of that insecurity. And the one thing that occurred to me, the example of this that has made headlines to such a great degree was Google's inadvertent sniffing of WiFi globally as they roamed around collecting their geolocation data for Google Maps and other geolocation services.

And so what Lance, for example, or anyone else could mention is that email is not secured; that even though the login may be secured, it is typically the case that with your typical email client the contents is going across the Internet and maybe in the air unsecured. And so, for example, if Google happened to be, or somebody else happened to be wandering around sampling what was in the air when this customer statement with this personal information was being retrieved by the customer, it could be sucked in, just in the same way as all the other personal information had been collected inadvertently by Google.

So, I mean, this is in that same classification of the kind of stuff that any third party monitoring would be able to pick up. I don't know if that'll make sense to somebody who refuses to understand that their service is not secure. But it is not something that's easy to understand.

**Leo:** Well, yeah. I mean, I'll be honest with you, I don't know how much of a breach of privacy having somebody see what your balance is. It would be one thing if they sent the credit card number through the mails.

**Steve:** Yeah.

**Leo:** I mean, I don't know. I get all my balances via email now.

**Steve:** It's funny, too, because there have been, I saw just recently someone who did not have an ecommerce capability who was taking credit card information sort of manually. Well, and they said, send your card in, like, four separate pieces of email. It's like, okay, well, that's better than all at once. But still, certainly not very secure.

**Leo:** Yeah. You know, I presume this is secure, but I just got a little dongle that you plug into your iPhone or Android phone into the audio piece, it's from a company called Square, SquareUp.com. It's a credit card reader. And I guess it turns it into audio. And then you have software on the phone. And without signing up for Merchant Services or anything, you just sign up for an account with them, and of course they take a cut. You can use credit cards like that. I could walk up to you, and you could swipe your card onto my iPhone or Android phone.

**Steve:** Yeah. You can imagine like in all kinds of, like, little trade shows or...

**Leo:** Exactly.

**Steve:** ...farmers markets...

**Leo:** Precisely.

**Steve:** ...sort of scenarios, yeah.

**Leo:** Jennifer, my wife told me about it. She said, "I was at the craft fair. Do you know about this thing?" she said. "Everybody's swiping credit cards." I said yeah, it's really - it's Jack Dorsey, the guy who started Twitter. It's very interesting. All right. We have three questions that are all kind of about the same, about the RFID stuff. So I'll just read them all at once.

**Steve:** Okay.

**Leo:** And then withhold your applause till the last one. Didier Stevens, our good friend Didier in Brussels, suggests an RFID tag in a wristband of a watch: Steve, I know someone who keeps his subcutaneous RFID tag lodged into the wristband of his watch. Then you don't need to inject it. He always has it with him. There's no surgery involved. That's one way to do it. Efrain in Miami, Florida thinks an RFID-enhanced cell phone might be an idea. I think rather than implanting a chip in our bodies, what about a chip in our cell phones? With the chip being in our cell phones, it can handle complex things because it's a powered device. Seems to be a logical choice. I think we can all agree that our phones are always within reach. And more likely for a company to give you a cell phone with a tracking chip than ask you to get it implanted surgically.

And then Eric in Palm Coast, Florida says, concerning RFID and having the public key advertised, I know this may be unlikely for most of us, but could you not be the trigger for your own assassination? While this may be an extreme example, could we not be targeted in many other less sinister ways as well? Additionally, much of what you thought would be cool was available to Bluetooth users a decade ago. You walk into a room, your music would resume, your Mac would unlock - yeah, that's from a Salling Clicker, using Bluetooth. Presumably a little less secure, though. And I've mentioned this NFC, Near Field Communications, in the new Google phone. Similar to that; right? As we talked about at the beginning.

**Steve:** Yeah. So certainly many people suggested alternatives to implantation, which I completely agree, installing something in your body is marginally radical. The point that I made last week was people are doing it. There are hobbyists who are on bulletin boards, actively talking about where the best location is to put the implant. One of our listeners has I think a wife or girlfriend who is an acupuncturist, who was concerned that the location that we've been talking about, sort of in the web of your hand between your thumb and your first finger, which is where it seems to be a popular location, that's an acupuncture point that's related to your upper intestines or something. So maybe that wouldn't be the best location. I mean, I don't know.

So I did like Eric's reminder about Bluetooth because, if you wanted to back away from installing, from implanting something in your body, all of our phones are Bluetooth-enabled. And it is certainly the case that you could turn Bluetooth on, on your phone; not have it discoverable so you don't have that concern. And then, in the scenarios like I was talking about, like being able to walk up to your garage door and just press a button and not have to use a key or a keypad, or just have your front door of your house unlocked whenever you're nearby.

Certainly a cell phone is becoming virtually ubiquitous for all of us. We've got one on us pretty much wherever we are. The downside is that it can be taken from you. You could lose it; or you could imagine somehow, if someone really wanted to get their hands on it, they could snatch it from you or something, where it's much less easy to do that with something embedded in your body. On the other hand, I mentioned last week, you wouldn't want to have someone cut this thing out of you violently if they wanted to...

**Leo:** If they knew it was there.

**Steve:** If they knew it was there, if they had some access to it. So I guess, yeah, certainly there are alternatives to embedding it. And the idea of just sticking it on your wristwatch I think is a good one. I talked about some silicon bands, like the...

**Leo:** I think a lot of people don't wear wristwatches anymore.

**Steve:** Exactly. As a matter of fact, I'm still a wristwatch wearer.

**Leo:** I am, too. But we're old.

**Steve:** Most of my friends are no longer using wristwatches. They're constantly checking their phone to see what time it is, if they're concerned. But they're just not wearing a wristwatch anymore. I think wristwatches are kind of becoming pass.

**Leo:** Well, they're jewelry now.

**Steve:** Old technology.

**Leo:** They're no longer functional; they're just jewelry.

**Steve:** Yeah.

**Leo:** Hmm. I remember there were some Mexican legislators, back when there were a lot of kidnappings going on in Mexico, who got RFID tags implanted, I guess in case their body should turn up somewhere and not be identified.

**Steve:** Tom, who did the podcast, both of his dogs have RFID tags.

**Leo:** So do our animals, yeah.

**Steve:** And I know that there's a nightclub in Brazil that tags its members. I guess if you join the nightclub, then you can be tagged.

**Leo:** That's funny.

**Steve:** And then walk in the back door or automatically get in or prove your membership that way.

**Leo:** Better than a password.

**Steve:** I have to say, though, I'm glad that Eric reminded me about Bluetooth. I could imagine that being an answer for me, although it'd be a little tough to hack my car. Be nice if my car knew me. But certainly a laptop. You can imagine adding something to a front door lock, or even a garage door, so that if this particular Bluetooth radio was, I mean, if it's got the right amount of range, I don't have to go invent new technology from scratch and so forth. So that's sort of a possibility.

**Leo:** I'm pretty sure Schlage, the lock company, makes a Bluetooth-enabled door lock.

**Steve:** Interesting. We'll check.

**Leo:** If it's RFID, anyway. If not Bluetooth, some sort of RF technology. Dustin B. in Seattle, Washington wonders about Controlling Bandwidth: Steve, I'm aware this isn't in regards to a previous episode. Therefore it's not feedback, per se. But I was pondering a question I felt you were the perfect person to ask. How do ISPs limit bandwidth to specific households? Hmm, that's an interesting question. I hear so much about digital, meaning everything is either on or off, no in between. So clearly the physical connection to my house isn't changing. I'm able to change my service speed with my provider without getting a new router. So what is Comcast doing when they say now you're 20mbs instead of 5mbs? It's the same connection. Thanks for all the podcasts. You guys started the same year I graduated high school and headed to college in '04, and it made me realize I needed to switch from a business degree to web applications.

You know, I was at Google a couple of nights ago for their media event, and I met several Googlers who said that they listened or watched shows that you and I did and others did, and that's how they got into technology. I met one guy, he said I was in politics, and I listened to TWiT, and I realized I loved technology more than politics, and now I'm working at Google. So we do make a little bit of an impact, Steve.

**Steve:** That's neat. Okay. So I mentioned Mark Thompson, my buddy in Phoenix, who's got an insane amount of bandwidth at his home. The way Comcast and others function, and this is sort of within the realm of cable modems, is that the coaxial cable itself has an insane bandwidth capacity. If you consider the idea that over a coax cable is flowing how many television channels? And a TV signal, I mean, even HD is an insane amount of data. And most of these have all switched to digital now so that this is digital data, an amazing amount of digital data flowing over this coax.

So essentially you can think of the pipe that's connected to the outside side of your cable modem as being just massive. I mean, it's a huge pipe that's capable of carrying a phenomenal amount of data. So all that Comcast or any other cable modem management provider has to do is tell the cable modem how much of that torrent of bandwidth to sip from, essentially. There are channels, and the state-of-the-art cable modems are able to be scaled in terms of sort of how many of those channels of data they're going to be sipping from at one time. And it can be scaled up to whatever the maximum data-handling capability of the cable modem is.

So even though, yes, you're receiving ones and zeroes, and there is some digital granularity to the rate of upload and download, because the pipe on the other side is so huge, this coaxial connection can potentially carry so much data, you just tell the device, the modem that's hooked to it, how much of that to take. And it's able to.

**Leo:** There's a widely used and well-known Linux program that's a proxy called Squid that is used to do this. And we used to help people set up Squid servers in their house because their roommates were sucking too much of their bandwidth. And for a while I think we were doing that here. I think that one of our routers, I think it was running the Tomato firmware, had that capability. And we had limited a router for our visitors to 900 kilobits so that they wouldn't kill our bandwidth. So that's a fairly easy and well-known application.

**Steve:** Yup.

**Leo:** We still do that.

**Steve:** There are, now, that's sort of different. There's, like, bandwidth shaping and bandwidth throttling. And there you may have a high-capacity single link where you're wanting to throttle the bandwidth of different...

**Leo:** Individuals.

**Steve:** ...users within the link. And, for example, you're doing it by IP or by port or whatever. And there it actually is a little tricky to get TCP, which is the protocol, or even trickier for UDP. Sometimes what they're actually doing is they're queuing the packets and allowing them to pile up a little bit because TCP will notice the delay in the connection and slow itself down as it notices that it's not getting acknowledgements back from the other side as quickly.

**Leo:** Oh, that's clever.

**Steve:** So it turns out that it actually is surprisingly complicated to throttle TCP connections in a smooth way. But it's a problem that's been solved.

**Leo:** Yeah. Let's talk about Chrome OS. Question 8, Ty, Nashville, Tennessee. He says: I love listening to the podcast. It's one of my greatest resources for technical learning and growth. I know it's a little early to tell, but what do you think of the security of Google's Chrome OS? I have the Google Cr-48 kind of demo laptop that they sent out right here with me. I've been playing with this. And actually I talked to some Google - the product manager about this very subject. He says: I know it's early, but what I've read makes it sound like it'll keep local storage to a minimum and only allow downloading of a small subject of file types, meaning it will not even be running executable code outside of the browser. I've even read it will monitor system files for changes on startup - that's right, it actually does a hash on the firmware and the system files, and if they're modified it goes, whoa - and repair them if it sees any modifications.

It almost sounds too good to be true from a security standpoint (not privacy, of course, since Google is running the show and users are required to log in with a Google ID to even use the system). And that's also true. Can you think of any obvious drawbacks to the platform, or have you heard of anything that would give you pause in giving it a try? Thanks for your work and passion for technology.

I talked a little bit, I mean, that's one of the main points of Chrome OS. Everything, for instance, they sandbox Flash. They sandbox the reader. Not everything is yet sandboxed, but that's their goal. So that's really what they're trying to do. Every tab is sandboxed.

**Steve:** He starts off saying he knows it's a little early to tell, but what do I think about it. What I think is, everyone who listens to the podcast knows that one of our fundamental lemmas of security is that it's not something that can be stated. It's only something that can be proven. And so it inherently takes time. What I love about this is that it's been designed with security awareness from the beginning, and I couldn't ask for more. So whereas Windows' legacy unfortunately, or the Internet's legacy unfortunately, predates security completely.

For example, where DNS - there was no concern, no thought about security back when the Internet's fundamental architecture was being created. So it's always had that problem. Similarly, Windows was designed as a single-user system that then became networked, then went on to be part of this global network and has suffered, well, as we just saw, in 2010 Microsoft broke their record of security vulnerabilities. Even now, years after they've been security aware, and Ballmer's been stomping around onstage proclaiming that their operating systems are the securest ones they've ever made, well, they just broke their record this year in number of security patches. So I couldn't be more pleased about this.

Now, I did also read, though, that there is technology, and I don't remember the name of it, but it allows Chrome to run native code. And it's like, oops, that sounds a little scary. Someone is pushing for more performance. And I'm hoping that it will be sufficiently sandboxed that running native code rather than scripting can be made safe. Scripting

being inherently interpreted, you potentially have more control over it than you do if you're running native. But also being scripted you've got a performance hit.

So I just - I love what Google is doing. I love the idea that because they had the advantage of starting so late on this, they've been able to - and because they know that nothing matters more to us than security of this, the idea of offering this as a secure platform is really compelling. So I've got my fingers crossed, and I'm holding my breath that it ends up being proven to be as secure as they have designed it to be. But the fact that they have designed it to be sure gives it a head start.

**Leo:** Yes. It's certainly their intent. And I love the fact that things like system files and firmware are hashed and protected from modification. I think that's, I mean, now you're really starting to pay some serious attention to this.

Number 9, Christiaan Conover in Annapolis, Maryland wonders where a one-time password model would work for RFID tags: I've been listening to your discussion on RFID tags which sound very intriguing. You did mention some security concerns, naturally. But the benefits and use cases of such technology do sound appealing. What I started to realize as I was listening is that some of the uses sound similar to how I use my YubiKey.

This got me thinking. Maybe the YubiKey model is exactly the solution to many of the concerns around RFID tags. It's already an open protocol and an authentication standard which can be implemented by anyone. So it would take care of the need for a standard to be developed. It could be set to issue a one-time password at each use so that somebody trying to clone your chip with a reader wouldn't get much benefit as all they'd get from the read was that single instance of OTP (one-time password). Plus it would give the user the ability to control authorization of use by requiring them to confirm a certain device or location to be allowed access and be able to revoke it at any time. You could easily send a text message or email to somebody when authorization is needed, which they could reply to and within seconds a new authorization rule could be created on the fly.

Is it possible to do that with the way RFID tags work? Or have I missed a technical detail that would preclude this? It seems like a proven secure authentication method and a natural choice for a technology like this. Thanks for a fantastic show. Avid listener. Wednesdays are now one of the more exciting days of my week. Take care. Christiaan. It seems like there's not enough computational juice in an RFID tag. Is there?

**Steve:** The problem actually is that a one-time password requires that some way of everyone knowing...

**Leo:** Synchronization, yes.

**Steve:** Yes. Some way for everyone knowing that you've used that password, and so it can't be used again. In my use model for why it would be intriguing to implant something, it's that I'd love my laptop to know me, my garage door, my car, my phone, different devices. It would be nice to have a physical proximity acknowledgment. But those are inherently, or at least in some cases - my front door - not on the network, not attached to a global network.

So for one time - the whole concept of a one-time password model, the YubiKey for example, is that when we're authenticating with the YubiKey, we are connecting, the device to which we're authenticating has a real-time connection back to a central server. And any potential device to which we want to authenticate has to have a real-time connection to a central server so that essentially what we really have in our YubiKey, or we would have in our RFID, would be a counter. That counter would be encrypted, and that would produce the one-time password. So the value in that counter is being maintained at the central server, so that the server knows what the last authentication was and is able to track that counter forward as it advances. That requires everything to which we would authenticate to be tied into that server.

So unfortunately, nice as it would be, that really doesn't fit the use model for a standalone authentication. Something like some sort of cryptographically enhanced RFID does. Unfortunately, I don't think it's a one-time password model.

**Leo:** You must have been talking about your beloved side tabs in Firefox with Tom. Nick in Thief River says Chrome has side tabs. Side tabs are experimental in Chrome. But if you type about:flags into Chrome you'll get a settings page where you can enable a Side Tabs context menu option in the tabs context menu. So, yes, you can do that in Chrome. Is that one of the reasons you don't use Chrome is because you like your side tabs?

**Steve:** Well, I'm in love with side tabs, and I've got, like, 75 of them open right now. And I'm not one of those people whose desktop has four icons. I don't know what it is. Something's wrong.

**Leo:** These are kind of cool settings. I did not know about this, about:flags.

**Steve:** They are. There's a number of cool settings there. There's one at the very bottom that I liked also.

**Leo:** Click on a blocked plug-in to run it. Web GL enables canvas events. GPU-accelerated compositing.

**Steve:** That's what it was. It was the other GPL technologies for speeding up CSS rendering stuff.

**Leo:** Wow. Enables 3D CSS and higher performance compositing of web pages using your GPU.

**Steve:** And so, for example, you made the comment, Leo, that your, as is mine, your wide screen on your most recent Mac...

**Leo:** Right, yeah, a 16:9 screen. And a lot of computers now have 16:9 screens,

have all this real estate horizontally, and often, like in the MacBook Air, are very constrained vertically.

**Steve:** Yes. And so you might try it.

**Leo:** I'm going to.

**Steve:** You enable the first one. Then go and right-click on a tab, and you'll see the last item on the tab's context menu says something about vertical tabs or side tabs or something.

**Leo:** So Enable Tab Overview, is that the one I should enable?

**Steve:** The very first one.

**Leo:** Okay. That's the first one. I've enabled that.

**Steve:** Okay. And so now, if you right-click on the tab - maybe you need to restart. Oh, yeah. Whenever you make a setting change, a button appears at the bottom that restarts Chrome.

**Leo:** Oh. I should have probably said "okay" or something. All right. Yeah, it's still enabled. So now what do I do? I do a tab.

**Steve:** Right-click on a tab. And the last item in the context menu should say something about side tabs. It did work for me, and it immediately moved the tabs over to the side. And it's like, oh, this is good.

**Leo:** That's neat, yeah. Well, I'll play with it a little bit more. It's not doing it for me, but I'll play with it.

**Steve:** Oh, it says Use Side Tabs, and mine has a checkmark on it now.

**Leo:** And does that put all the tabs - oh, yeah.

**Steve:** It creates a nice column over on the left-hand side. It gives you, as you would like, more vertical real estate by moving them off. And I have to say, every time I look at Chrome, I think, wow, this is just the cleanest UI.

**Leo:** They've done a nice job. I am a Chrome, exclusively now, Chrome user.

**Steve:** Seems pretty quick, too, Chrome does.

**Leo:** Oh, yeah. Snappy. Nathan Ramsey from Australia, now in London, has some very nice things to say: Steve, after studying off and on for a couple of months, I just passed my Security+ exam today. All I can say is it was a tiresome yawn. That's not a negative on Security+, but a positive on Security Now!. Listening to you week after week has imbued me with the ability to understand words like "honeypot," "least privilege," "DNS spoofability," et cetera, with the greatest of ease. I was amazed at how much everything you've told us follows - sometimes word for word - with the best practices that I had to study for this exam. It almost felt like you wrote the exam. That's pretty neat.

Anyway, I just wanted to thank you and Leo for your devotion to such a technically useful show and your commitment to provide nothing short of the best. That's really, I think, an outstanding trademark of Steve. Absolutely committed to perfection. Kind regards, Nathan (living in London, UK but from Australia where I started listening to you). Well, thank you, Nathan.

**Steve:** I just wanted - I just ran across it. I thought, well, that is just the neatest thing, that he took his security exam, and it was like, okay, yeah, I know all this already.

**Leo:** I hope you passed it, Nathan.

**Steve:** Oh, he did.

**Leo:** Oh, good. Now, ladies and gentlemen, if you will, the Adobe Gotcha! Tip of the Week. Steve, I really enjoyed your Security Now! podcast, says Jack D. of Port Perry, Ontario, Canada, and want to thank you and Leo for a superb job. Let me pass along something I noticed that your listeners should look out for. When I recently updated Adobe Reader from 9.4.1 to the new version X - which is a big "X," Mac style - unlike previous updates, I suppose because this is a new version number, it reenabled - oh, boy.

**Steve:** Mm-hmm.

**Leo:** It reenabled JavaScript and reenabled "Allow opening of non-PDF file attachments with external applications." These are the things you said to turn off because you don't need them in a PDF reader, and they're a huge security threat.

**Steve:** Yup.

**Leo:** So I think all of our listeners and viewers by now have that turned off in Reader. But it turns it back on when you update.

**Steve:** Yup.

**Leo:** I'm unsure whether allowing these settings is no longer a security threat under this new sandboxed version. But I thought I should point it out. You know, even if it isn't a security threat, turn it off.

**Steve:** Exactly. Again, one of our other fundamental security rules is, if you don't need it, and you're not using it, turn it off.

**Leo:** And most people don't.

**Steve:** Features are bad.

**Leo:** Yeah, features are bad. I like that.

**Steve:** Features are bad.

**Leo:** Turn it off. You don't need it. Well, thank you, Jack D., for pointing that out. I haven't - I don't use Adobe Reader, so I have no idea.

**Steve:** Yup, and it caught me by surprise. I checked, and it's like, ooh.

**Leo:** It did the same thing?

**Steve:** Yep, absolutely, those things were back on. It's like, oh, you bad people.

**Leo:** That's not good.

**Steve:** They just don't get it.

**Leo:** Yeah, no kidding. I mean...

**Steve:** They really don't.

**Leo:** Very few people need JavaScript or external third-party programs running from a PDF. That's just not - that's an unusual usage.

**Steve:** Yeah. I mean, and all they are is big, huge security vulnerability opportunities. It's like, get those things turned off. So I thought all of our listeners would want to know that, had they gone up to v10, to go back into their properties, down to trust management, and remanage their trust.

**Leo:** Turn off, yeah, remanage your trust. Turn off JavaScript and third-party apps.

**Steve:** Yup.

**Leo:** Steve, as usual, a great show. Thanks to Tom Merritt for filling in last week. Next week, we don't know. Oh, you have an idea of what we're going to cover? Or is it a mystery?

**Steve:** Don't have any idea.

**Leo:** No idea.

**Steve:** There are some things brewing here that may end up grabbing us.

**Leo:** We can talk about that OpenBSD story.

**Steve:** Some big news, yeah.

**Leo:** If you want to know more, go to GRC.com/securitynow. Steve has show notes there. He has 16KB versions for the bandwidth impaired, full transcripts - thank you, Steve, for paying for those and getting those done. GRC.com. You know, while you're at GRC, browse around because not only is there a ton of free stuff there, like the DNS Benchmark, ShieldsUP!, Shoot The Messenger, DCOMbobulator, all of those great programs Steve has written in little tiny teeny-weeny assembly code, there's also his bread and butter, the No. 1 hard drive maintenance utility in the world, SpinRite. If you have hard drives, you need SpinRite. And you can get it from GRC.com.

Watch us do this show every Wednesday, 11:00 a.m. Pacific, 2:00 p.m. Eastern time at live.twit.tv. And if you miss the live broadcast, don't worry, we make it available in audio and video at TWiT.tv/sn or on iTunes, on Zune, anywhere you can get - where finer podcasts are found. Thank you, Steverino. I shall see you next week.

**Steve:** A pleasure, as always, and see you next week.

**Leo:** Thanks, on Security Now!.