
# Listener Feedback #124

**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-314.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-314-lq.mp3

---

**Leo Laporte:** This is Security Now! with Steve Gibson, Episode 3.14159, recorded August 17th, 2011: Your questions, Steve's answers, #124.

It's time for Security Now!, the show that protects you and your friends online from predators and bad guys and hackers.

**Steve Gibson:** Those, too, yes.

**Leo:** And here's the man in charge of catching that predator - actually, no, you don't catch them, you just expose them - Mr. Steve Gibson of GRC.com, the Gibson Research Corporation, manufacturer of SpinRite, the world's - when you're talking software, I guess you're not a manufacturer, you're a creator.

**Steve:** Publisher, yeah.

**Leo:** Also lots of freebies and, of course, the host of Security Now! for 314 episodes, more than five years now.

**Steve:** Yeah, well, more. Actually, we crossed into - two weeks ago we did 312, which is six times 52. Or, wait, yeah, six times 52. We're now in year seven, Leo.

**Leo:** What the heck? More than six years. Wow.

**Steve:** More than six years, yeah.

**Leo:** We didn't - we didn't note that. But, wow, that's great. Congratulations.

**Steve:** Yeah, and welcome back, by the way, for all of our regular podcast listeners. They know that Tom was hosting for two weeks, and you're done with the trial, and so...

**Leo:** Jury duty is over. And that's where the predator came in. It was an online child predator case which got thrown out for entrapment. But a very - boy, it was interesting. The whole thing happened five years ago, and it was at a time when people were really scared of the Internet. And there was this perception that the Internet was loaded with predators. We've learned since that it's not. But at the time people were terrified. And I think that's when this happened; you know?

**Steve:** Yeah, that's a real good point. I hadn't thought about that what a difference five years makes in Internet time is, like, amazing. We should also mention for our listeners who are used to receiving the podcast on Wednesday evening or Thursday morning, that next week I'll be recording, I'm switching places with Paul again. He's going to take the Wednesday slot, and I'm going to take the Thursday slot. So Windows Weekly will be early, and Security Now! will be a day late.

**Leo:** That's great.

**Steve:** But not a dollar short.

**Leo:** Thank you for doing that. We appreciate your flexibility. We try not to do that, but Paul's doing a lot of traveling lately. So we have a Q&A #124 in our continuing series of your questions, Steve's answers. We're going to get to those in just a second. But first let's get some security updates.

**Steve:** Okay. So lots of Mozilla news. Something that I was really glad to see was the announcement that version 7, believe it or not, we were just at 3, then 4, then 5, now 6 is available. We're soon to have 7, late next month, late in September. Version 7, among other changes, will be shrinking Firefox's memory appetite by 20 to 30 percent. There's a two-month-old project called MemShrink, which began in mid-June, which has been working on finding and closing memory holes and leaks. One of the developers was quoted saying "Firefox v7 uses less memory than v6 (and 4 and 5), often 20 to 30 percent less, and sometimes as much as 50 percent less. This means that Firefox 7 is faster, sometimes drastically so, and less likely to crash, particularly if you have many websites open at once and/or keep Firefox running for a long time between restarts."

Well, this is something that I have noticed because I watch memory consumption pretty much constantly. I have, like, Task Manager open on one part of one of my screens just because I'm seeing this problem. And I've noticed that I'm still on 5, and it's been saying that 6 is available, except that 6 is incompatible with three add-ons that I would like to be using, so I'm holding off jumping up to 6 until they catch up. But if I close Firefox, then I see memory drop like 500 megs, half a gig. And then, if I restart it, all the same

tabs reload. And I should say that's about 50 tabs that I have open. I love tabs. And then as the tabs load, memory creeps back up again, but only to about maybe half to two thirds of what it was before. And I have noticed also that, if I close tabs, that memory is never recovered.

So I'm really delighted that Firefox is - that the Mozilla guys are going to be addressing this in Firefox because I've been looking enviously over at Chrome, thinking, well, it isn't really quite as tab-friendly as Firefox is, and I really like my add-ons. Ghostery is there, but NoScript, as easy as it is over Firefox, is not yet available over on Chrome in the same fashion. So I'm staying with Firefox for now.

**Leo:** So funny, we just were talking about Firefox 4 coming out, then 5 came out. And now 6 has come out. I guess we should stop looking at numbers; right?

**Steve:** Well, and in fact Mozilla is going to be helping us with that. They're taking the version numbers off of Firefox. They've said that they're going to be removing Firefox's version numbering from view completely. It will not be in the "About" window anymore. The official position is that users don't care about version numbers.

**Leo:** Yeah, but we care - well, wait. We care if the plug-ins don't work.

**Steve:** That's a very good point.

**Leo:** I mean, Chrome doesn't trumpet version numbers, but you can always go and look and see what version number you're using.

**Steve:** Right. So what's happening is, Asa Dotzler, who's one of the developers, he announced in a Bugzilla bug report that the Firefox "About" dialogue will stop listing version numbers, but simply state that the browser checked for an update recently, and that the user is running the latest version. Those who want to know what version they're running can consult the "about:support" window.

**Leo:** Oh, okay.

**Steve:** So you have to type "about:support" into the URL, and that'll then bring you lots of details about stuff. So, I mean, I think we're seeing them being influenced by the Chrome model, which is - actually I had a lot of people tweeting that they got a kick out of my saying that the Chrome version numbers reminded me of the U.S. debt clock because it was just going all the time, and the number just kept changing. Every time you looked, it was different. So a change is coming for add-ons also in the so-called "Aurora" release, which is available I think now in pre-beta. If you put "Firefox Aurora" into Google, it'll take you to a URL where you can download.

What they're doing is they're going to start protecting us from software which adds add-ons to Firefox without our explicit knowledge and permission. We've often talked, for example, about how you can get toolbars and add-ons added unless you're very careful installing things. And in some cases you'll add software that won't even ask for your

permission or knowledge. So what's going to happen is that, starting with Aurora, which is not yet in general release, but it should be soon, especially at the rate that these guys are going, you will be asked to explicitly authorize any add-ons that have appeared in Firefox, but you have not yet authorized, when you upgrade to a new version, like a major version.

And I'm not sure how they're going to handle this not showing you the version versus major version. We talked about this a few months ago, saying that you would always be given the opportunity of acknowledging a major version update. So maybe that'll still be the case. You just won't see it in the About box. You'll have to go to this about:support URL page in order to get that. So I guess that's what's going to happen. So anyway, so it's good news that anything that is installed without our explicit acknowledgment will not be run. Firefox will say, okay, here's something new. You need to give the browser permission to run this. And presumably you could say, whoa, that's not something that I want, and just decline to accept that, which is, I think, really good.

So there's been some controversy over how many Flash bugs Adobe has fixed recently. Our friend Tavis Ormandy, who's a security researcher at Google, blogged that he had given Adobe 400, reported 400 Flash bugs, although Adobe was counting them as 80. Adobe's Brad Arkin, who is the senior director of product security and privacy, posted to Adobe's blog, as a consequence of some of this back-and-forth, he said, "There's been some chatter about CVE numbers lately, so I thought it would be helpful to clarify Adobe's position on how we use CVEs to communicate product security information. CVE.mitre.org describes them as 'international in scope and free for public use, CVE is a dictionary of publicly known information about security vulnerabilities and exposures.'"

So Brad continues: "Unfortunately" - and I have to say he brings up some good points here. He says, "Unfortunately, there are many differences in opinion on how CVEs should be used in real-world situations. If there are four instances of unsafe buffer usage resolved with a single buffer size check, does that represent four CVEs or just one? If vulnerable code is copied and pasted into multiple products, should the vulnerable line of code be described with a single CVE or one unique CVE for each product? How does the answer change, if the product is vulnerable because of a linked vulnerable library rather than copied-and-pasted code?" And he says, "The real-world questions go on and on."

**Leo:** Yeah, but in the real world we just care about vulnerabilities. We don't care if it's cut-and-pasted. It's a vulnerability.

**Steve:** Just fix the damn stuff. So I got a kick out of SANS Institute, one of the SANS Institute editors, William Murray, was quoted in a recent SANS newsletter about this controversy. And he said, "Instead of disputing the number, Adobe should be trying to identify what they are doing fundamentally wrong. Whether there are 400 or 80, there must be a pattern here somewhere."

**Leo:** Well, I don't know about that. The pattern is ineptness.

**Steve:** Now, I have to thank Glenn Frasier from tweeting this morning something that just hit the news because I could see me, SGgrc, getting deluged with people making sure I had seen this and wanting to know what it meant. So I wanted to let everyone - I wanted to get ahead of this one this time. So thanks, Glenn. AES is still safe.

**Leo:** Whew.

**Steve:** Two years ago in May we reported on what's called a "related key attack" where some very sharp security researchers had discovered that there was a slight vulnerability in what's called the "key scheduling" portion of AES. And I covered key scheduling for AES in great detail on a podcast we did specifically about how AES works, where we completely dissect it and look at it. So if anyone's curious, you can go back and find that.

As has happened repeatedly, and happened again, attacks against crypto things, as Bruce Schneier has famously said, "They only get better. They never get worse." So what's just in the news today, which Glenn picked up on and forwarded to me, is an advancement of this such that it no longer requires four related keys. It basically is a weakness in a single key which has a four-to-one strength-reducing effect on AES key strength. So, for example, it takes a 128-bit key, and since it's a four-to-one reduction, that's two bits. So it has the effect of reducing a 128-bit AES key to 126 bits. Well, that's significant, but we still have vastly more protection than we need. And AES can run 256-bit keys, which aren't affected at all.

But it brings some real-world scale to this. Even with this new attack, so-called "attack," which Schneier has already said doesn't really matter, because Bruce has weighed in, he said - no, this is not he. But the effort to recover a key is still huge. So, for example, the number of steps to find the key for AES 128 is, in decimal, it's an 8 followed by 37 zeroes. So to put this into perspective, on a trillion machines that each could test a billion keys per second, it would still take more than two billion years to recover an AES 128-bit key. So note that large corporations are believed to have millions of machines, and current machines can only test 10 million keys per second.

So there is no company or organization that we know of that has that many machines that are nearly that fast. And even if they did, we would have to wait two billion years for one single AES-128 key to get broken. So that's, yes, it used to be eight billion years; now it's two billion years. So it's worth watching, but I wanted to put everyone's mind at rest. AES has not cracked. some But two bits' worth of weakness was found. And that's significant; but we still have, as far as we know, the strongest cipher around.

Oh, and this is actually out of place, but something about Firefox has been driving me batty. And I just discovered what was going on this morning. And I wanted to let any of our listeners know. We've talked before about how nice it is to be able to float your mouse over a window and use the scrolling wheel in order to scroll that window without having to click on it to give it current focus, which I think is just super handy. And in fact one of my favorite little pieces of freeware is called Kat something, I want to say KatMouse. Yeah, K-a-t-M-o-u-s-e. So when I updated to Firefox 5, it sort of stopped working reliably, which, again, I just can't stand. I love having the inertial scroll for, like, moving through web pages.

What I discovered this morning is that it is scrolling the wrong tab. I had opened one of Tavis Ormandy's PowerPoint presentations, or maybe it's a PDF; but it's, like, paged. And for some reason, as I was closing tabs, I noticed that it was on a different page than I had left it. And so by just experimenting this morning, I realized that successive pages, when they weren't being scrolled by KatMouse visibly, it was a different tab that was receiving the scroll message. So, Mozilla, if you guys are listening, I hope you found this or fixed it in version 6. And if not today in version 6, then in version 7 probably by tomorrow morning. So that would be great because it would be nice to have that fixed.

And Leo, you and I talked when iPhone 4 came out about the question of whether Gizmodo was going to be in hot water for publishing that. So I thought I would just mention that prosecutors in California have decided that they will not file charge against the tech blog Gizmodo for its purchase of an iPhone 4 prototype which they bought. And then I guess Jason Chen, an editor at Gizmodo, was going to be potentially be in some hot water for showing us all the iPhone 4, which of course annoyed Steve Jobs to no end. The guys that sold it to him are still in trouble, probably.

**Leo:** Oh, yeah. They got prosecuted. In fact, I think they got convicted.

**Steve:** Ooh. And a San Mateo County assistant district attorney said that the difficulty his office faced in terms of prosecuting Chen was that Chen and Gizmodo were primarily, in their view, engaged in a journalistic effort to conduct an investigation into the phone, so they were protected by the editorial shield law.

**Leo:** Good, thank you.

**Steve:** Yes. That was, I think, the right outcome.

**Leo:** Yeah, I agree. And I think it was the right outcome to prosecute the guys who stole the phone. I think that was also the right thing to do.

**Steve:** And then profited by selling it.

**Leo:** Yes, exactly. That was a little sleazy, yeah.

**Steve:** I wanted to acknowledge somebody, unfortunately it flashed by my Twitter feed so I can't give credit where it's due. Someone mentioned that Lion fixed one of my biggest pet peeves about the Mac, which is you can now drag any window border. You no longer need to separately move the title bar and then resize in the lower right-hand corner. It's like, oh, thank you. So for anyone who misses that feature in Mac OS, we have it now. Which is really great. And, Leo, when I was up there sitting next to Kevin, you and Kevin were talking about how Lion changed the scrolling direction so that it was more like a touch screen, so it was taking you guys some time to get used to it. And I refused to put up with that. So...

**Leo:** Of course you did. But you could change it.

**Steve:** Yes. And I wanted to make sure everyone knew. I just went into Preferences, and sure enough, right there is a checkbox saying "return it to the proper way of scrolling."

**Leo:** If you're using, as I am, a trackpad, it actually does make sense. You get used to it pretty darn quickly. But I can't say I blame you. And certainly if you're using a

scroll wheel mouse, it doesn't make any sense. You really do want to change the setting.

**Steve:** And I didn't have a chance when we were up there doing TWiT. So is it the idea of, like, dragging the page rather than dragging the scroll bar?

**Leo:** Precisely. So when I move my fingers up on the track, it moves the page up. When I move my fingers down on the track, it moves the page down. And that's kind of direct drive. It certainly makes sense when you're touching the screen. There's no question on a pad that's what you should do. The issue is it's not exactly direct drive when you're using a trackpad. That's 50-50. When you get to a mouse, and you're using a scroll wheel, scrolling down moves the page in the completely opposite way of how you expect it to go. But, you know, gaming has always had this problem. When you do a flight simulator, or even just use a look-around in a game...

**Steve:** Push your stick forward.

**Leo:** ...there's two different ways to do it. And it's completely analogous to this situation. And it's different strokes for different folks. Some people change that, invert the mouse look, and some people use the standard mouse look. And you just do what you like.

**Steve:** You just said "different strokes for different folks." Which of course is a great pun.

**Leo:** It's perfect.

**Steve:** It is. Okay. And last week we covered - we did the second in our series on How the Internet Works while you were in trial, Leo, discussing ICMP and UDP. And I got a bunch - I didn't realize, I mean, I'd heard them years ago, but I just wanted to share them with people. A bunch of people tweeted me some old longstanding jokes about, of all things, I mean, this is serious geek jokeness. Kevin Panko tweeted, he said, "I would tell you a UDP joke, but you might not get it."

**Leo:** Now, that's good. That is the geekiest joke. I mean, only geeks would get that; right? Or not, as the case may be.

**Steve:** Exactly. I would tell you a UDP joke, but you might not get it. And there are about four or five variations on that theme. So I wanted to thank everybody who sent those to me because I got a kick out of it.

**Leo:** That's really good.

**Steve:** And then Matthew Stinar tweeted, he said, "Steve, hearing you talk about UDP

being unreliable reminded me of a joke I read: 'The best thing about TCP jokes is you always get them.'"

**Leo:** Even if you have to retransmit once or twice.

**Steve:** One way or the other, you're going to get the joke.

**Leo:** Love it.

**Steve:** Yes. And I wanted to tell people my plan is next week when I'm up there in person with you, Leo, I am going to unveil this multi-month project I've been working on. What I have been working on, and I did mention it this week with Tom, I've come up with a paper-based cipher, a strong encryption that uses nothing but a piece of paper.

**Leo:** This is, by the way, the holy grail for amateur cryptologists. They're always looking for ways to do non-computer based because crypto's been solved really for computers, but non-computer-based crypto.

**Steve:** Well, and so this is called "Off the Grid" because it is - it's non-technology. It uses no computers. And it is based on a specially constructed grid which you use as a reference. And I began working on this immediately after publishing the Password Haystacks page because the one thing we still don't have is a way of doing per-website passwords where you don't need to write something down or memorize something, but you still have a different password for every website. This encrypts the domain name into a secure password in a way that cannot be reversed. And so I'm going to tell everybody about a really interesting journey I had in developing it, and all about how it works, next week when I'm there in the studio with you.

**Leo:** Web32 in our chatroom said, "Oh, Perfect Papyrus Passwords." Actually, you know, Neal Stephenson in "Cryptonomicon" came up with - and I don't know, I've never really seen it vetted, but a playing card-based crypto system. And I seem to remember that he worked with somebody like Bruce Schneier to validate it as a working crypto system. So it's very interesting. And you still do need it because there's spies and other things, people who don't have computers, I guess.

**Steve:** Well, and the problem, of course, is that if it's in your computer, it is susceptible to being compromised. I mean, we had a scare, as we all know, with LastPass a few months ago, them worrying that their database had gotten loose. There's some concern that the solutions that hash that are running in your browser could expose your master password. So there's a certain group of people who would just like to have something low tech. And so this is low tech, but high security. And so I think, I know it's going to be a great episode. The thing that made me think about it is I want to next talk about TCP, the TCP protocol. We can't do that next week, so that'll be three weeks from now, after the Q&A between the Off the Grid crypto system and then talking about TCP.

I did want to share a really neat story that's titled "SpinRite Saves My Teeth." Christian Alexandrov, who's in Sofia City, Bulgaria - and we can tell that English is not his native

language, but his is much better than my Bulgarian. So he says, "SpinRite did it again. This time SpinRite saved my teeth. Hello, Steve. I decided I want to share this story with SpinRite followers. I survived a car crash recently caused by drunk driver speeding and having difficulties controlling his BMW. This driver hit the car behind mine, and it hit mine. After all ended, I had some injuries on my head, nothing serious. But the most of my injuries required dental help. At this time I was unable to afford such serious interaction from a dentist in terms of money, but I had no choice. Health is more important.

"A friend of mine is a dentist who has the equipment to heal me and the qualifications to do so. I was about to ask him to give me time to pay him with few separate payments in time. He told me he cannot do much. His computer works poorly and cannot access all patient records, including my dental records."

**Leo:** I see where we're going here.

**Steve:** You see where this is going. I told him I can fix this as soon as pain is relieved to be able to focus on the task at hand. He gave me painkilling injection and allowed me to fix his PC. I booted SpinRite on Level 4 to check his drive. We got to the problem late. The drive was in deep trouble already. But we got lucky because it was not too late. We quickly saw green 'R'" - meaning recovered - "icons when SpinRite began its work. SpinRite found two areas that were bad. One contained one Windows XP system DLL file needed for the OS to work at all. The second area contained pieces of the database with the dental records. While the dentist was trying to focus on my first healing session, I had firm belief in SpinRite. After two days of constant work" - presumably on the hard drive and not his mouth...

**Leo:** I hope.

**Steve:** "...SpinRite completed the hard drive maintenance reporting that it successfully recovered all damaged areas and marked them as unusable for the drive. On the drive map I saw two green "R" icons, which was what I was hoping for. I repeated the process. On the map I saw two "B" icons on the places where the green "R" icons were before, indicating these areas will not be used again. After the second pass on Level 4, SpinRite says that there were no bad areas. Great. We rebooted the PC, and jaws hit the floor." Which I thought was an interesting pun in this case. "The system booted so very fast, and the database with all dental records was there in perfect condition, opened in a second and responding very fast to inputs and searches. The dentist was so happy that he asked me for advice, how to prevent this. I told him to buy two NAS servers and use them as redundant backup for his date."

**Leo:** There you go.

**Steve:** Yeah. "I told him two in case one dies, to have another copy for precaution. The dentist promised me he will heal me for free. Then, a few healing sessions later, we made an arrangement. I will maintain and fix his computers. In return, he will heal me every time I need dental help. I have four more healing sessions until I'm fully healed and fully recovered. Thank you, Steve, for this great piece of software, and thank you Steve and Leo for this great podcast. I wish best of luck to GRC.com and TWiT.tv from a

happy SpinRite user."

**Leo:** That's great.

**Steve:** So I just loved...

**Leo:** That's six or seven sessions, we're talking 10 grand, 20 grand? That's a lot of work.

**Steve:** Boy. And he must have been in pain because he said he needed an injection in order to even be able to think straight. So thank you for the report, Christian.

**Leo:** Don't forget to floss. All right, Steve. I've been perusing these questions. There's some good ones in here. Are you ready?

**Steve:** You betcha.

**Leo:** All right. Question #1 comes from Stuart Henry on Twitter, @quotingstu, a winemaker and web developer just up the road apiece in Napa, California. He says: Do ICMP packets have a TTL value? You'd better define that one for us.

**Steve:** Well, okay. So this sort of hails back to last week's episode and three weeks ago, the first two episodes in our How the Internet Works series which we are firing up. And the answer is yes, because everything that the Internet carries is enclosed in an IP packet. So remember that the IP packet is sort of the - we used the analogy three weeks ago, you'll remember this, Leo, of the Russian dolls, where you've got a smaller one inside a larger one inside a larger one inside a larger one. Or you could also think of it as an envelope inside an envelope inside an envelope.

Well, the outer envelope is the IP container which has the - and this was the brilliance of the Internet designers - only the things, the minimum necessary information for getting the packet from the source IP to the destination IP. So it contains fields for each of those. But, for example, there's no port numbers there because that's not about two different machines. That's about more of the content of the packet, which on this level of the hierarchy we don't care about. We only want it to get from A to B.

But we also need the minimum amount of information for that job. And one of those is the TTL value because we need packets not to make the mistake of living forever. They have to die. If for some reason there's a router loop that is sending them in a big circle, we want those packets to expire, as we discussed last week. So, yes. Everything - ICMP, UDP, TCP, all the other protocols, because they are contained inside the IP envelope, inherit all the characteristics of that containing envelope, one of which is a TTL. So that's fundamental to a packet on the Internet, just like the source IP and destination IP.

**Leo:** Cool. I'm sorry I missed the first two of those episodes. I'll have to go back and

listen.

**Steve:** Yeah. Actually last week. I've had a huge amount of positive feedback from people. I guess I'm just, I don't know, I'm in the zone for these. And we're developing sort of a style that a lot of people like, saying that, gee, they thought they knew how the Internet worked until they heard this, and now they've really got it, now really get it. So...

**Leo:** I really love that is the beginning stuff. In fact, somebody sent me an email saying - this is really funny. Just today I got this email saying, "You should do more help and how-to. I'd really like to know more about like how operating systems work, how the Internet works." And I said, "Well, check out - I think people, because it says "Security Now!" think, oh, I'm not interested in security. But you've done so much great basic education and how not only security like crypto works, but how computers in general work. This is a...

**Steve:** Well, remember we did that whole series on the fundamentals of how computers operate.

**Leo:** From ground zero all the way up. And I think that that - you couldn't get much better than that. Mike Kowalczyk with Question #2 in Montgomery, New Jersey, he's wondering about IP stack vulnerabilities. Steve, as a long-time listener I want to thank you for the great show and for your insightful coverage of all things security. I'm enjoying your series on How the Internet Works. In last week's episode 313, ICMP and UDP, you discussed the IP stack that resides on all routers and Internet-connected devices. My question is, what is there to prevent malware from modifying the IP stack? Just one possible attack that came to mind would be to modify the stack on a router or computer to always decrement the TTL, the Time To Live, to zero so that no packets are ever delivered. How is the IP stack implemented, or what's in place to prevent this from happening? Oh, boy. Thanks, and keep up the great work. Mike.

**Steve:** Okay. So I haven't really - we've used the term "stack," but I've never explicitly defined it. I'll probably...

**Leo:** It's not the kind of stack that we talk about in memory, a memory stack; right?

**Steve:** Correct. It is not.

**Leo:** Okay. That's confusing.

**Steve:** Yeah, it is. And I will discuss it again, not in a Q&A, just because it's an important concept. But it's worth mentioning here. It's a conceptual stack which mirrors the hierarchy of the Internet's packets. For example, the normal term is a TCP/IP stack. And the idea is that you have the application that wants to communicate, like your browser or

your email client or your DNS client. And it sends its data into the operating system, which conceptually has organized these layers in a stack, a stack of layers.

So, for example, the data that the web server is going to use uses TCP. So that data gets put into a TCP packet, meaning that it's wrapped with the TCP headers. Then it goes, it drops to the next layer down in the stack, where it was just the IP layer. And so whatever comes from above gets wrapped in an IP layer. Then it goes to the next layer, which is the physical layer, the Ethernet layer, which wraps it in an Ethernet packet, and then off it goes out of the machine. Packets coming back in get the reverse treatment. First the Ethernet wrapper is taken off. It's checked and taken off and then passed up the stack to the next layer, which is the IP layer. The stack examines that, makes sure that it all looks right, checksums are balanced, it's the correct IP address and so forth. If so, it removes the IP wrapper, essentially sort of opens the envelope and takes out what's in that layer of the envelope and then passes it up the stack to the next layer, which is going to be UDP or TCP or ICMP, whatever it is that was wrapped in the IP layer.

So that's where this stack concept comes from. It's a stack that sort of represents a hierarchy of protocols. So as we move down, we're wrapping things in layer after layer, like these Russian dolls. And as we go up the tack, we're reversing the process, taking successive wrappers off until we finally get up to just the raw data that we actually wanted to send to a remote point.

So Mike's question has a little bit of confusion here because he refers to routers and computers as being synonymous in this way, and they're not. Routers don't have stacks. They process IP packets, but you can talk to a router where it is the endpoint of your discussion. For example, you can telnet to a router using the telnet protocol over TCP. And there it's a TCP endpoint; you're connecting with it.

But the actual routing process doesn't involve a stack. The router takes in an IP packet, looks to see where it's going, and then decrements the TTL as we just discussed in the first question, and as Mike references, if that Time To Live hits zero, the router says, well, this packet is dead. And so it doesn't forward it. Instead it sends a note back to the sender saying, hey, by the way, this packet never got to its destination. It died before it got to its final proper destination.

But routers themselves in the routing aspect of them don't have stacks. The IP stack involves interpreting the protocol, and routers really don't do that. They just receive packets. They don't care what it contains. They look at where it's going. They make sure it's valid to forward it, and then they send it out over whatever link is closest to where it's going based on the routing table, which we'll be talking about in a future episode.

So Mike's question, with that background, is what's to prevent the stacks or, for example, the software in the router from being altered so that they don't work? And the answer is, well, only the same kind of protections we normally have. For example, the system's stack is located down in the kernel, where software is not supposed to be able to get to it at all, and the kernel works to protect itself. So it's trying not to allow the OS to be altered, and the stack is part of the OS. So those same protections pertain. But remember that any software could, rather than, like, setting the TTL to zero so that the packet would die, it could just not - the router could just say, I'm not going to send the packet. So there are all kinds of things that could break that would cause routing not to work that are actually even simpler than setting the TTL to zero. Just choosing not to route it would do the same thing.

**Leo:** I mean, I think there are a lot of hack attacks that involve hacking the TCP stack.

**Steve:** Oh, yeah.

**Leo:** Yeah. That's a very common attack. In fact, sometimes you know that happened because you'll remove spyware, and you'll lose Internet access. And that's because the spyware was embedded in the TCP/IP stack.

**Steve:** Yup. There's something called an LSP, a Layered Service Provider, which is a protocol stack in Windows. And traditionally spyware has installed itself in there. And the way it works is you have to sort of knit everybody together. And if you just yank that out, you just lose all your connectivity with the Internet.

**Leo:** Jared in Western Australia has a question for you about traceroute. He says: What determines the exact path taken by packets? When you do a traceroute, does the path taken always remain the same, or does it change from moment to moment?

**Steve:** Well, it's a great question. And part of the original genius of the guys that designed packet switching was they realized it doesn't matter. The idea is you have this vast network of routers that are interconnected. And you inject a packet somewhere, anywhere in this single global, I mean, it became global now, network, where it's got a destination IP. And the job of any router receiving that packet is to forward it on in what it thinks is the best direction. This system is as robust as it is because, if a link goes down between two routers, well, then the router goes, oh, I don't have that link, which used to be my preferred link for that packet. So I'm going to send it out of this link.

So packets, they're sort of like little autonomous bits of information that have a destination address, and they're just going to try to find their way to that address one way or the other. So it's often the case when, like, something is flaky, there's something called "router flapping," where an interface is flapping, which is the term used for kind of coming up and down. It looks good for a second, then something goes wrong, and the link is broken, and then it comes up again, and then it goes down again. And so network engineers call that "flapping." So if you were, like, doing a traceroute over a path that happened to have a flaky link like that, then you might find that from one moment to another the actual hops being taken along the way would vary. And there's really nothing wrong with that. Everything on the Internet works because there really is no notion of the proper path. There's just we want the packets to get there.

And what's really interesting, we'll be talking about this in TCP, is that ICMP and UDP, which are just sort of raw best-effort protocols, there's nothing to say that the packets even arrive in the same order. You could send three packets out and have, because of interface congestion or a link that goes down, you might have those three packets take different routes, running over links of different speed, and they could arrive in a different order. So obviously you don't want files being transferred having their blocks of, like their buffers of data arriving and, like, being reordered in a different sequence. So TCP deals with things like out-of-sequence packet ordering; whereas UDP and ICMP that we discussed last week don't. So the whole network works, even if the actual specific path taken varies from one moment to the next, which is just so cool.

**Leo:** Question #4, Bill in Michigan asks and reminds: Not ICMP but IDENT. In 313, Steve, last show, you were trying to remember how an IRC server was checking if you were there. I'm sure you were thinking of IDENT, that's TCP port 113, a port that's usually not stealthed - in fact, I know you know this, Steve, because this is a whole big thing about whether an IDENT port should be open or not - and not ICMP, which you were talking about at the time. I think that's what ZoneAlarm has had as an adaptive reply. This IDENT check was also used on SMTP and FTP. To this day, routers may be stealth for every port except port 113, as they try to reduce service calls, since by not replying to port 113, sending mail was causing problems if this IDENT handshake was timing out. I think that's mostly older clients, but I might be wrong. As you recall, you would usually respond with your host and username in the old days of a standard UNIX environment. Hope this helps, but you probably have a zillion of these by now. Bill in Michigan, and who's a regular in our chatroom as bill_mi.

**Steve:** Yeah, he's actually a regular contributor over on GRC's news groups.

**Leo:** He's great, I love Bill, yeah.

**Steve:** Yeah. So thank you, Bill. What happened last week, Leo, is that we were discussing ICMP, and I just - it had been so long since I had thought about this, I thought, you know, I remembered that there was something that was this. And I couldn't remember, I sort of knew that I wasn't sure. But anyway, so Bill did pick up on it. And I wanted to close this little link for our listeners, that it was that port. And one of the things that I liked about ZoneAlarm was it was the only firewall at the time that did this adaptively. If unsolicited probes to port 113 came in, ZoneAlarm had them as stealth. It would drop the packets.

But if you were in the process of establishing a connection, for example, to a remote SMTP server, and the SMTP server pinged you, essentially, by trying to connect to your IDENT port, then ZoneAlarm was smart enough to say, oh, this is coming in from an IP that we're in the process of trying to connect to, so do allow that because we don't want to upset that remote server by having it wonder whether there's really anybody home or not. So, yeah, so thank you, Bill.

**Leo:** Yeah. You knew. This is one of those things that Steve has known for so long that it just got - it happens to me all the time now. There's stuff, like, of course I knew that. You knew that because you've dealt with it directly with ShieldsUP! all, I mean, this was a big issue with ShieldsUP!.

**Steve:** Yup.

**Leo:** Adam Gilman in Minneapolis, Minnesota has questions about how the Internet works. Steve, let me first off start by staying I'm fairly new to Security Now!. Well, welcome, Adam. I've only been watching for a few months now and was first turned on to TWiT by Kevin Rose when watching Diggnation. Thank you, Kevin. By the way, just a note, Kevin just tweeted this morning that his dad passed away, so our

condolences to Kevin Rose. And I've met his dad, and he was a great guy and was very proud of him.

**Steve:** Couldn't have been very old.

**Leo:** I don't think he was. Maybe early 60s at the oldest. So probably a bit of a shock. So Kevin, our sympathies and condolences to you. Adam continues: I'm loving your How the Internet Works segment. Most others like this are very high level and boring, but yours is fascinating. You see, we don't do high level here. This show we do down and - this is the assembly language of podcasts - down to the metal. In the last episode you were talking about packet TTL and routers responding to dead packets. It made me think, wouldn't this be a problem and perhaps a source for DDoS attacks? If an attacker were to send out packets to random destinations with TTLs less than the necessary to reach the destination, and then spoof their source IP so it's the same as the attacker's target machine IP - so it looked like it was sending itself, I guess, a packet.

**Steve:** It would be the IP that the attacker wanted to attack.

**Leo:** Oh, oh, oh, I get it. So you'd get all these NACKs. Wouldn't that turn all the Internet routers into bots generating mass amounts of ICMP Time Exceeded packets - I've got to do the page turn here - towards the attacker's target? I get it. I get it. Wouldn't this be a problem as the packets are being generated by the Internet routers doing, are doing what they're supposed to do by sending all these ICMP Time Exceeded packets back towards the source IP that the attacker spoofed in the original packets? I'm sure that they've got route - well, let Steve answer this. Hopefully this makes sense. I'm just getting into Internet security. I'm not sure if I'm missing something. But what do you say?

**Steve:** He is absolutely right. And it turns out that we went through a period of time, maybe, what, 10 years ago, where there were lots of small botnets that were being run by script kiddies. And it was sort of early technology. And this was one of the many ways to create bounced traffic is you send - you could certainly send a packet out that would expire. But you could also just ping. Ping generally generates an echo. So you could just, for example, ping routers or destinations. Any remote IP that will respond to a ping, you could spoof the source IP so that it looked like it was coming from your attacker. And you'd get in trouble.

The other thing you could do was you could send a - you could try to establish a TCP connection. So you would send a TCP SYN packet to any web server that would send a SYN ACK - and we'll be discussing what all this is when we talk about TCP in detail - send a SYN ACK back to the apparent source. But if you spoof the source IP, then it would actually go to your victim.

So what happened was it looked like all these different websites were attacking you. So, I mean, this is an example of one of the many ways that bad guys have found over the evolution of the Internet to get up to some mischief. And it absolutely does hide the attacker's identity. It requires backtracking to figure out where these things are really coming from. And you can't use the source IP. You have to literally follow the traffic from

one router to the next in order to come back. So, yes, I mean, that's a perfect example of one of the ways that this beautiful technology can and has been abused over and over and over.

**Leo:** Question #6, Robert in Pasadena, Texas - I didn't know there was a Pasadena, in Texas - comments that our sci-fi recommendations are excellent. Steve, I just have to say I recently took your recommendations for the Lost Fleet series, and I love it. I'm halfway through the sixth book now, and I only started a week and a half ago. Wow. Do you have recommendations for books of similar style? I could never get into the Peter F. Hamilton books, but I fell in love with the Lost Fleet series. Who writes the Lost Fleet series?

**Steve:** Okay. So, oh, shoot.

**Leo:** Should I look it up?

**Steve:** If you Google it, you'll find it immediately.

**Leo:** Is that sci-fi? Is that our friend from Sci Fi - Arizona?

**Steve:** No, no, no. This is something that I found because I was looking for series, book series.

**Leo:** Jack Campbell.

**Steve:** Yes, exactly. And I talked about these. What I loved about these was this is like big, like military space fleet movements, but in real detail, where the constraints of speed of light and speed of acceleration and deceleration are mixed into the plot so that this commander who is actually, he was in suspended animation for a generation, during which time a huge sort of civil war among human colonies got underway. And all of the people who knew real fleet movement got killed. And so he gets found in a life support capsule, and this is - I'm not giving away much because you get this in the first few pages of the first book. And he is resuscitated. And it turns out that technically he's got seniority because...

**Leo:** He's an old guy.

**Steve:** Exactly. Even though he's still young, 150 years ago he was, like, in charge of things. And so he remembers how to fight space battles with knowledge that has been lost from the human race. And I have to say, I mean, I really enjoyed these. So I did want to let people know about the Lost Fleet series. But I can sort of understand what Robert means about the Hamilton books. And I love the Hamilton books. I think we wrap up our Q&A with someone who agrees with me. And the next thing I'm going to be reading, as soon as I finish with FreedomTM, will be I'm going to plow into the Galactic Center, no, that's not - I just finished those. The Hamilton - help me out. You just read

those, Leo. We were talking about it.

**Leo:** Oh, the Void Trilogy.

**Steve:** The Void Trilogy.

**Leo:** Oh, man, is that good.

**Steve:** But I do have another series that I really enjoyed that's sort of more along the lines of the Lost Fleet. A little easier, a little breezier, I mean, here Robert, he's been reading, he's in the sixth book after a week and a half.

**Leo:** Yeah. That's pretty breezy. You couldn't do that with Hamilton.

**Steve:** No. And so, Robert, slow down because we're going to run out of books for you if you keep reading at this pace.

**Leo:** Although I'm looking on Campbell's web page, and he says that next Lost Fleet manuscript was just turned in.

**Steve:** There's going to be another one? Yay.

**Leo:** Yeah, it's Lost Fleet: Beyond the Frontier novel "Invincible." Not till May. But it is coming out. And he is in town right now, or he's in Reno, anyway, at the World SF and Fantasy Convention going on right now in Reno, Nevada, so...

**Steve:** Well, the Lost Fleet series, you could read one, and if it grabbed you, you could know that you've got five more following it because they're really interesting. So the other series that I really liked also, along the same line, it's not like Peter F. Hamilton grade, but it's called Helfort's War, and it's a series by Graham Sharp Paul. And Michael Helfort is a newly minted academy graduate, set in the future, who happens to be good at stuff. And I will leave it there.

Now, the one annoying thing is we've got the first four books, but it turns out - and I thought that the books one through four was it. But book four was sort of the half. And all the other books finished off so that you weren't left hanging. Book four ended with a cliffhanger. And now I'm waiting for book five, and I'm really looking forward to it. So I really enjoyed these. But be warned that the series is not complete. And in the case of the Lost Fleet series, I had to wait for book number six because - but it was worth waiting for to see how he wrapped it up. And now I guess there's going to be a number seven. So I'm delighted for that.

**Leo:** Well, it's all stylistic. I mean, I think I'm not that - in fact, whenever Peter F.

Hamilton starts doing space battles, I just lose - I can't follow it. And so if you like that kind of thing, then obviously the Lost Fleet series would be great for you. It's just different - what's nice is that, even within the genre of science fiction, there are many, many subgenres. And whatever you like, there's something for everybody. So that's great.

**Steve:** Right.

**Leo:** I agree. I appreciate the recommendations. Always looking for something new to read.

**Steve:** And so for what it's worth, if you like the Lost Fleet, Helfort's War sort of has a similar feel to it. And the lord knows Robert will be done by the day after tomorrow.

**Leo:** Yeah, at the rate he's reading.

**Steve:** With those four books.

**Leo:** Question #7, David Taylor, Atlanta, Georgia. He shares some comments about the ICMP/Ping security issues you talked about: I've been in networking and network security for quite a few years, and the Internet backgrounders have been a nice, nostalgic reflection on how things got started. From my paranoid security background I thought I'd add a few interesting points to some of the comments you made in last week's podcast. You mentioned about the ICMP echo request/reply being filtered these days for security reasons. One of the reasons you didn't mention was one that I always considered a pretty neat idea - and a strong reason to block ICMP in and out of a corporate network.

Someone quite a few years ago came up with the notion of using the ping request/reply traffic as a tunneling mechanism. ICMP echo reply/request packets can be any size that - this is wild - can be any size you like - I've lost it, it's the page turns that kill me - from the minimum packet size up to the max path that's allowed by the MTU. The payload is generally random data or a chunk of unallocated memory thrown in to pad out the packets - data leaks, anyone? - and is commonly ignored. Why don't they just put zeroes in there? I don't know.

I suspect you've probably seen the trick as you've been around the block a few times. But just in case this one snuck by, someone decided to use the payload to hide encrypted data. This made for a pretty sneaky back channel for controlling a machine that's been hacked. You send them a ping, in that little ping, and then there's this whole rest of the packet that's got encrypted data in it. Many sites block incoming traffic, but not outgoing traffic. With this in mind, the firewalls in many places, especially at home, will happily let these ping packets pass all day long unnoticed, containing your passwords or whatever else. Just thought I'd pass this one along. Dave Taylor, CISSP. Wow.

**Steve:** Yeah, I did want to just sort of acknowledge that. Actually in my spec for my

VPN, using ICMP as one means of establishing a connection is already in my notes. So I'm well aware that ICMP is a general purpose and very capable data carrier. So it's absolutely the case. And as I was mentioning last week, normally when you get an echo reply, the reply packet contains the echo request so that the recipient can see what it was that it sent out that generated that response. And so you are able to put whatever data. Sometimes I've seen just like, you know, the alphabet, ABCDEFG and so forth. Or sometimes, as David says, no one even bothers to initialize the contents of the package. They just say, eh, we'll just send out whatever's there because it doesn't really matter.

**Leo:** You just get random stuff.

**Steve:** Yeah, it actually is a payload-carrying protocol.

**Leo:** Wow. Question #8, Jeff Hornung in Indianapolis admonishes us that corporate IT is not so bad. And, yes, encryption is tough. First off, I love our broadcast. I've listened to them all. Of course I'm a SpinRite owner. But in the past I've been somewhat dismayed by the negative comments directed toward corporate IT and suggestions on how to get around the controls that we as corporate IT professionals must put in place. We want to do the right thing to protect our customers and our employees and of course comply with HIPAA and GLBA and PCI and DOI and many other regulations. I think you two have great experience and speak with some authority. But I don't think either of you have experience in corporate IT for today's finance industries. And I think we'll both cop to that. That's right.

**Steve:** Yup.

**Leo:** So if I could give you some tidbits from my perspective: Recently you mentioned there is no overhead to decrypting all this data. While I'd love to see everything encrypted all the time, it actually is a very big deal. Some people responded to your comments but missed a major issue. My company has UNIX, Linux, Windows Servers, IBM zSeries mainframe, IBM iSeries, that's AS/400 platforms. The hundreds of applications that we use must communicate across all the platforms and systems, internal subnets and firewalls and databases. They perform thousands of lookups and file transfers daily. To have all of these using fully encrypted data all the time is a lot more overhead than you are giving credit for. CPU, processing translation - going from probably big Endian to little Endian and ASCII to EBCDIC and things like that - and key management are significant and time consuming. Keep up the great work, but please spend some time getting to know our situations.

**Steve:** Well, for what it's worth, I really never meant my comments to refer to Intranet encryption, but rather Internet encryption. So the idea being - and I guess I wasn't clear about this, Jeff. And to any other listeners I apologize. When I'm talking about the need for encryption, we've talked about the need to encrypt databases, which I think is crucial. And so that's not about network encryption, that's about encrypting the data at rest while it's at rest so that, if you lose control of it, all it is is pseudorandom noise. And then, where I'd like to see encryption 24/7, 365, and forever, is when it's moving out across the public Internet, where remote servers would be using SSL and TLS certificates in order to establish and enforce encryption.

So I really understand what Jeff is saying, that internally there'd just be a huge amount of overhead for very little return, for example, managing certificates that are expiring on all sorts of machines. And I can imagine you just pull your hair out. But it's really, hopefully, you've got firewalls and border security such that you can have trusted information inside your corporate Intranet and then deal with encryption for communications outside on the Internet.

**Leo:** All right, here we go, Question #9 from Andy Williams - I always loved his music - in Philadelphia, Pennsylvania. He wonders about a good CPU for projects: Steve, I've been thinking about really getting into some toy OS development, and I was wondering what CPU you thought would be the best. It could be real or emulated, old or new. I'd love to have your professional opinion. Thank you, Andy. You were going to do, when you retire someday, you've got all those PDP-8s here behind you. You were going to do a PDP-8 operating system because that's a simple processor for something like that.

**Steve:** Yeah. I have to say, having written a bunch of code for it, as I did in order to make those lights flash that way and so forth, I'm thinking that it's just too annoying to write to a 12-bit instruction set.

**Leo:** I agree with you.

**Steve:** Yeah, the thing's got - it's got a 3-bit opcode, so a totally of eight instructions. And I spent all my time - basically, the accumulator, it's got one accumulator and a multiplier quotient registry. It was just a pain in the butt. So I think what I want to do is something on instructions that I really enjoy. But I did recently survey what was going on with development platforms for, like, small processors for the infamous Portable Sound Blaster project. And I found something that I really like a lot. There of course is the Arduino project that I know you and Andy have talked about. Arduino is an interpreter that runs on a number of small, embeddable processors. The problem is, for me, and I think for our questioner and listener Andy, it probably isn't the right solution because he's talking about toy OS development.

What I found is a fantastic little board. It's only 29.95, that is, $29.95. It uses a state-of-the-art ARM Cortex-M3 processor. This Cortex-M3 processor has 512k of program flash memory, 64k of RAM. In the processor hardware, brought out to pins, is a 10/100 Ethernet; USB 2.0 host, device, and on-the-fly USB port; two CAN buses, which is the industrial bus used often in cars; four UARTs, Universal Asynchronous Receiver Transmitters; three I^2 C buses, two SSP buses, and an I^2 bus; an 8-channel, 12-bit analog-digital converter; 10-bit digital-analog converter; seven pulse width modulators; and on and on and on. It does a 32 by 32 multiply in a single cycle.

It is a state-of-the-art CORTEX M3 ARM processor for 30 bucks. And the entire tool chain, the development tool chain for Windows, Mac, and Linux is free for this thing. So $30 is your out-of-pocket cost. It's got a USB port that you plug into the machine and then just go crazy. And the FreeRTOS operating system is available for it. So you could start with something on an existing little free real-time operating system platform and then just start playing. So it's what I'm going to be using as the heart of the acoustic experiments that I'll be doing. And I would recommend it for anyone who just sort of wants to play with something. And you can program it in C, C++, or in its own native thumb language.

**Leo:** Neat. That's a fun thing to do, I think.

**Steve:** Oh, yeah.

**Leo:** You know, I wish life were longer. There's so many cool and fun...

**Steve:** Oh, Leo, gosh. I wish I were 12 right now.

**Leo:** Yeah, no kidding. No kidding, all the options. And I have to say, you know, we set up the ham shack in the corner, and it's so cool. But one of the hams I was talking to, we're getting a lot of noise, and there's a ton of RF noise in the environment these days. You know we're downtown, so it could be PG&E transformers, it could be neon signs, it could be the 802.11 in this building, it could be our fluorescent lights. There's so much RF in the environment now. And one of the hams was saying, oh, you should have seen it in the '50s. It was quiet. It was calm. Because none of this stuff was out there right then.

**Steve:** Interesting. Interesting.

**Leo:** In some ways it might have been better to be kids when we were kids because there were fewer choices. You could build a Portable Dog Killer, you could do what you did, or you could maybe be a ham. Now there's an infinite amount of choices. It's kind of overwhelming, isn't it.

**Steve:** Yeah.

**Leo:** It's so much fun. And I would love to, I mean, I would love to do this - I'd like to do some Arduino projects. I want to do more ham stuff. I mean, it just goes on and on and on.

Here's an email from H.K., Shanghai. He's an expat living in China. A little view on Telex. He says: Being an expat in China, I smile whenever you refer to trusting the Hong Kong Post Office. Whoever they are, we say. But onto the meat of the matter: When you mentioned Telex last week I was quite excited. There are many ways to get around the Chinese Firewall, the Great Firewall of China. VPN will work; proxies a lot of people use. But Telex, believe it or not, is a new one. It would be cool to set up the Telex application, but I don't have any idea how to do it, and it's not at all clear from the website. Could you help? Any help would be appreciated. Telex would be helpful to get websites like Google+, Facebook, Twitter, IMDB, YouTube - all these blocked in China. Hulu, too. P.S.: With an average network speed of 140 kilo - I guess he's saying kilobytes per second, GRC.com loads faster than Google.com in Shanghai, China. You know, I have to - just a little plug for hams. The stuff that hams are doing with data, like PSK31, Telex, RTTY, this is exactly what hams do. And China and Asia in general is full of hams, amateur radio operators.

**Steve:** Interesting. And hard to block that with a corporate...

**Leo:** Hard to block, but easy to track, unfortunately.

**Steve:** Ah, good point. You are emitting radiation, yes. So what he's referring to, Leo, is something we talked about during one of the two episodes you missed. There's something called Telex.cc is - I think that's this website. But I wanted to let H.K. and anyone else who was wondering know that as far as I know, this isn't yet actually online. This was a paper that a bunch of - that some security researchers put together that talked about an extremely clever way of intercepting TCP connection setup, that is, SSL secure setup, where the nature of the key being negotiated would contain information sort of in an encrypted fashion so that it provided additional information that allowed somebody outside of the block perimeter to reroute your traffic. I don't think it exists in a ready-to-go fashion. I think so far it's just a technical capabilities paper that said, hey, this would work, so keep your eyes open. So certainly, if we hear more about, I will talk about it on the podcast.

**Leo:** Really interesting idea. There's always Morse code, the original digital.

**Steve:** Exactly.

**Leo:** Tom Corwine in New York, New York, has some additional information on GoDaddy and their EV SSL Certs: Steve, it was nice that GoDaddy wrote to correct you on their pricing, and that you chose to air it and share it. But there's one important thing they didn't say. 99 bucks for an EV SSL cert is only the price you pay when you first buy that. After that, 250 bucks a year. I had a client for whom I purchased an EV SSL cert from GoDaddy. After the first year was up, the client was automatically charged 250 bucks for the renewal. I called GoDaddy to ask what was going on, and the GoDaddy rep said $99 for the first year, 250 for the renewal. I asked why shouldn't I just buy a new one instead of renewing the current one. He said, "Sure, if you want to go through the validation process again." This made me so mad. I felt like I was being extorted for money and vowed never to do business with GoDaddy again. The $99 price is a total misrepresentation. It's a teaser price. Thought you and the listeners should all know.

**Steve:** Yup. So you and I talked about how upset I was when GoDaddy, without my authorization, tried to reauthorize, tried to renew a certificate that I had allowed to expire. Nowhere on their site was I able to go back and tell them, no, I don't want it expired. And then they complained when they couldn't authorize my card because, fortunately, I had used one of those PayPal one-use credit card numbers to buy the cert with GoDaddy. I was just playing around with that PayPal service. And sure enough, it saved me. So no thanks, GoDaddy. I'm going to DigiCert.

**Leo:** Yeah. I still am moving, I'm in the process of moving our registrations to Hover from GoDaddy, only because it's so difficult. But we don't use SSL on any of our sites. Maybe we will. Maybe we should at some point.

Finally, Philip Hofstetter in Zurich, Switzerland writes: Peter F. Hamilton - a huge thank you! He says: Hi, Steve. After hearing you and Leo praise Peter F. Hamilton's work so many times last December - sorry about that - I listened to "Fallen Dragon" and was completely blown away. But now, when you brought him up again about four weeks ago in the podcast, I finally accepted that I will hopelessly fall behind listening to your show, and started in with "Reality Dysfunction" in its audiobook version. You cannot - he says this in all caps. YOU CANNOT POSSIBLY IMAGINE how much fun I'm having. This is by far the most enjoyable kind of entertainment I have ever consumed. I haven't read this yet. I've got to get this book. This book is so mind-blowingly good, I can't even find words to describe it. Thank you, thank you, thank you for the time you took to tell us about these masterpieces. Philip. Wow. I'm a Peter F. Hamilton fan, but this is the original one we were talking about a couple of weeks ago, that long...

**Steve:** Wait, you have not read it?

**Leo:** No.

**Steve:** Oh, Leo.

**Leo:** I've got to get it?

**Steve:** Yeah. I mean, yeah, you've got time.

**Leo:** Plenty of time, Steve. You know what I do, I get them on audio books, then I just drive around the block a lot.

**Steve:** The Reality Dysfunction series, I think it's four, yes, it's four huge hardbacks or eight paperbacks.

**Leo:** Oh, great.

**Steve:** And it went on a little bit long for me. So, I mean, but if it's in audiobook format, maybe - oh, no. Robert said in Texas that he - he's the guy that reads a book every four hours. So these would slow him down. But...

**Leo:** I don't think - it's unfortunate, I don't think it is on Audible, unfortunately. Though some of the older Peter F. Hamilton - they just started putting Peter F. Hamilton on Audible, so.

**Steve:** Yeah. Anyway, for our listeners who have not yet discovered Peter, I really - I recommend "Fallen Dragon." It's a standalone novel, really good, gives you a good introduction to Hamilton's style. And then, boy, you're in for plenty of reading, if you

enjoy his stuff. And I did mention that some of his older stuff would be coming out in print that had been out of print. And I'll try to keep an eye on - let people know when that's available. The, shoot, I can't think of the guy's name now, he's sort of a spy with ESP powers. Greg Mandel, that's the guy, the Greg Mandel series is going to be coming out from Peter.

**Leo:** Apparently "Reality Dysfunction" is on Kindle. So that's good. I can do it on Kindle. I can't bring myself to carry around a big, heavy book anymore.

**Steve:** Nah. It'll be out on Audible before long.

**Leo:** It sure should. I'm excited, wow. Actually the entire "Reality Dysfunction" on Kindle is $9.99. I don't know, is that all of them?

**Steve:** Nice.

**Leo:** That would be great. Wouldn't that be great if that were the whole thing? Oh, no. It looks it's just the first one, The Night's Dawn. Oh, boy. Oh, boy.

**Steve:** The Night's Dawn Trilogy.

**Leo:** Well, then I've read the Night's Dawn Trilogy. This isn't the Night's Dawn Trilogy, is it?

**Steve:** Yeah, it is.

**Leo:** Oh, well, I've read that.

**Steve:** It's also known as - I was sure you had, Leo. It's also called "Reality Dysfunction."

**Leo:** Ah. I didn't know it was the same name. Yeah, I've read the Night's Dawn Trilogy. Yeah, thank god I'm done with that.

**Steve:** And of course - I know, I know.

**Leo:** Okay, yeah, very good, but, you know - yeah, yeah, I've read that one. Okay, now I...

**Steve:** And like I said, it goes on a little bit long.

[Talking simultaneously]

**Leo:** As does this show. But before we go, I just want to say, I want to thank the folks at TriCaster. You know, a lot of what we do here in the new studio is courtesy partners who helped us out. And NewTek has been so great. This TriCaster 850 Extreme that we use is absolutely the next step up from the TriCaster we were using in the old Cottage. HD, we get so much, yeah, look at all this, so many capabilities. You know who's really become an expert on it, Alex and Chad, the young folks here, are really good at using this. And we haven't even started to scratch the surface. We're still doing the static lower thirds. We can do motion graphics in the lower thirds, we could do all sorts of stuff.

And you'll see, including wild transitions, you'll see as time goes by, as we use more and more of the features of this, what an amazing device this is. We - like that. That's the reality dysfunction. We really want to thank our friends at NewTek and encourage anybody who's looking to do live video switching or video production to try the TriCaster series: NewTek.com. We don't do the green screen stuff, the virtual sets. There's so much in here that we don't use. And yet we're already having so much fun with it.

Thank you, Steve Gibson. Gibson Research Corporation, his company, is GRC.com. That's where you'll get a copy of SpinRite, the world's greatest hard drive maintenance and recovery utility. You also can get a copy of all of his free stuff like Wizmo, Don't Shoot The Messenger, DCOMbobulator. And this show, as well, 16KB as well as 64KB versions of the audio; full transcriptions, which Steve pays for himself, so I thank you for that, Steve. And do the forums there. They're great security forums. And of course if you have a question for Steve for our next Q&A episode, just go to GRC.com/feedback.

**Steve:** Thank you, Leo.

**Leo:** Next week, more of the Internet series?

**Steve:** No, next week I'm with you in your studio. We're recording on Thursday. And I'm going to take our listeners through my development, from the beginning, of a paper-based encryption system.

**Leo:** Oh, that's perfect because we'll probably have to show stuff; right? So we'll need to do it here so we can put stuff on the table.

**Steve:** It turned out very well.

**Leo:** Perfect timing.

**Steve:** So that's going to be a really great episode, I think.

**Leo:** So don't forget, Steve will be live in-studio, but it's not the usual day: Thursday, 11:00 a.m. Pacific, 2:00 p.m. Eastern, 1800 UTC at live.twit.tv, if you want to watch. We usually have room for maybe 20 or 30 people in-studio. We welcome visitors. If you'd like to see Steve, meet him, and watch as we do this live, email my sister Eva at TWiT.tv, eva@twit.tv, and we'll accommodate as many people as we can fit in here. We always have a few people in here, but we can get - I think we can get - we had 50 in here during opening week. And I know there'll be a lot of people who want to come see Steve. Steve, thanks so much.

**Steve:** Thanks, Leo.

**Leo:** Catch you next time on Security Now!.