Listener Feedback #123

Description: Steve and Tom 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-312.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-312-lq.mp3

TOM MERRITT: This is Security Now! with Steve Gibson, Episode 312, recorded August 3rd, 2011: Your questions, Steve's answers, #123.

It's time for Security Now!, the show that helps you stay safe online. I'm Tom Merritt, filling in for Leo Laporte, who's at jury duty, trying to stay securely off a jury. But joining us, the star of our show, Mr. Security, the man behind GRC.com, ShieldsUP!, SpinRite and more, Mr. Steve Gibson. A pleasure to be talking with you again, Steve.

Steve Gibson: Likewise, Tom. I got to see you when I was up last Sunday for the flagship mission of recording TWiT with the gang. So that was fun. And it's great that we have you to help out when Leo is otherwise occupied.

TOM: It was great to see that you actually have legs. You're an entire human being.

Steve: Yup, they're all here.

TOM: Yeah, a torso below the waist. So, good. And you're coming up for the party, as well; right?

Steve: Yeah, I'll be back.

TOM: Excellent. I can't wait for that. We've got some great stories to talk about today, and a question-and-answer episode.

Steve: Yup, this is, well, this is actually an important episode also. I realized, wait a minute, 312, why does that number seem special? Well, six times 52 is 312. So this is the last episode of our sixth year of Security Now!.

TOM: No kidding.

Steve: Yeah. And our 123rd Q&A that we've adopted to give our listeners a chance to have their voices heard and provide some feedback and ask questions and just sort of close the loop, being social, as we all are these days. So we've got another great podcast, I think.

TOM: Well, I am honored to be on 312. And you've been doing Security Now! longer than TechTV lasted.

Steve: Yeah, that's true, yeah. And the finances are even more sturdy than TechTV's were. So, yeah.

TOM: For sure. Let's get into the security news. Now, there is a - we talked about this in the pre-show. There's news going around from McAfee putting out a press release about something called Shadow Rat. We're not going to talk a lot about that, though, because it's just, as you put it, Steve, the news is, wow, hacking still happens.

Steve: Yeah, I mean, I saw the news. I thought, okay, what is this? And I expected to see some big new deal. And then I realized, wait a minute, this is not a big new deal of any sort. McAfee's basically saying, oh, for the past five years there has been an organization, and they do believe it's one group, that have penetrated a number of different companies and entities of various sorts. It's like, okay, and this is news how? And so we have talked about RAT before on the podcast, the Remote Access Trojan tool that is often used, that people get infected with.

And in fact there was some - it may have been that that was believed to be implicated in the now very famous RSA break-in, that that was the tool that was used. Some unwitting support person opened a document and got themselves infected, and that gave them a foothold into RSA's network. But it's like, okay, so this has been going on for five years. It felt more like McAfee just sort of trying to generate some news for themselves. So I thought, well - I didn't have it on my list. So we certainly have covered it, in any event.

TOM: Exactly. We've probably over-covered it. Our next story, water is wet. No, our next story is about a new anti-censorship system.

Steve: It's really interesting. A number of people tweeted me about Telex.cc. And, I don't know, maybe I'm getting jaded or something. I just thought, okay, what now? I guess I'm not too interested generally in anti-censorship. It's a little bit off the map. But when I saw what these guys did, what they had done, suddenly I got very interested. So here's the deal. They talked about how TOR, which we've talked about, the onion routing technology, which securely encrypts hops from one onion router to the next in a way that no router in the chain is able to determine anything but where it needs to send the packet on to. And so it decrypts in a series of layers to really provide extremely good anonymity.

TOM: That's why it's called The Onion Router, because of those layers. That's the metaphor there.

Steve: Precisely. However, it provides that, but it does not hide the fact that you're using it. So it doesn't provide you anonymity in that sense. It only provides you security against eavesdropping. But, okay. So here's the problem with censorship in general is that, for example, to take China, if you're in Beijing, YouTube is blocked for you. So you're unable to access YouTube and any of a number of different services that the

government doesn't want you to have access to. So of course services like TOR, which are identifiable, are also blocked.

So a group of very smart people, who I commend, are presenting a paper at this month's 20th USENIX Security Symposium to describe the technology that they've come up with. And they talk about it as, instead of an end-to-end system, which is the way, for example, SSL works, where we're providing endpoint-to-endpoint encryption, they've designed something they call an end-to-middle proxy that has no IP address.

TOM: Now, see, this is the interesting point because I thought The Onion Router was able to obfuscate your IP address; because of all those layers, because of all those hops, it would be impossible to track you down. But this paper is saying, no, they're acknowledging you can find out that there is an IP address there; is that right?

Steve: Well, kind of. You're right about what The Onion Router does. But the point is you need to connect to an Onion Router node at your end in order to send the onion of encrypted layers into The Onion Router network. So a censor can simply know the IP addresses of the available Onion Router nodes and block them from you.

TOM: Ah, so it's not your IP address, but they can get you because they know you're communicating with The Onion Router. I got you.

Steve: Precisely. Exactly. So DNS can be used for blocking. Hard IP addresses can be used for blocking. So these guys said, okay. We want somebody within a censored environment to be able to communicate to anyone outside of the censored environment with it being absolutely undetectable. And so that's what got my attention. It's like, okay, wait a minute. What have they done? And what they've done...

TOM: How do you get rid of the IP address? That's crazy.

Steve: How do you do that? What they've done is so clever. They call it, as I was saying, end-to-middle proxying. So the idea is that ISPs, that is, traffic carriers outside of the controlled environment, are participating. And again, when I initially read their Telex.cc sort of, they have like a cartoon-y introduction to what this does, they talked about that they were going to encode headers in the traffic that would reroute the packets on their way out. And I thought, I mean, okay, wait a minute. Headers, that won't work because headers are inside of an SSL connection.

So what they're saying is that, within a censored environment, there are services you cannot access. But there are still services you can access over SSL because you'd have to be able to do that because all password handling, you at least bring up a secure SSL connection when you're logging into a remote site with a password. So they're not blocking all SSL, and they're not blocking all sites over SSL. They're just blocking the ones that they want to censor you from.

So the idea is that, using this client - you get a client. You download it. A friend gives it to you. They recognize access to the client may be restricted. So they may literally be back to sneakernet in order for people to hand the special client software around from person to person within this censored environment. You use this client, and it establishes a standard SSL connection to a non-censored site. What this clever - and it is really clever - technology does is it monitors the establishment of the SSL connection, that is, and we've covered SSL extensively in the past on the podcast.

One of the things that happens when you're bringing up the encrypted link between the

two endpoints is that the client generates a random number which it will be using for its session. And we've also often talked about the need for randomness in cryptography. You need to be able to generate high-quality, unpredictable random numbers in order to create tokens which are only used once for encrypting during that session. What these guys realized is, if you were to monitor the establishment of the SSL connection, and the client was not using truly random numbers, that is, they were unpredictable, but they were deterministic, that is, somebody who had a matching software who was actually acting as a man in the middle - so this would be someone carrying traffic outside of that censored environment that was deliberately placed in a man-in-the-middle position - could watch the establishment of the SSL handshake, check the random number being provided by the client, and see if it matches specific cryptographic criteria; and, if so, recognize that this is actually a request for a redirection.

TOM: So it's not actually random anymore?

Steve: Well, it's random. You can sort of think of it as there are so many more bits in that random token than we actually need...

TOM: I see.

Steve: ...that there are ways of applying some constraints on it so that, if somebody had, like, so if somebody had a list of alternative destinations, they could see whether the client was secretly trying to signal that that was where they wanted this traffic to go. And so on the fly - but at the same time it's still random, it still provides security, and it still is a real SSL connection. It's just a way of almost steganographically embedding additional information in an otherwise completely valid handshake. And what's so cool is it is undetectable by the censors. They can't see that this is anything other than a standard SSL connection being established like millions are doing constantly. Instead, somebody outside of that environment can detect that and redirect the packet to basically just change the destination IP on the packet and send it to where the user inside really wants it to go. And it works. It's just extremely cool.

TOM: And there's no way for the censor to be able to tell it's got the extra information in it? They can't just also detect, like, hey, there's something wonky, the same way that the Telex station detects it?

Steve: Apparently not. I didn't go into the paper in detail. I do provide a link here in our show notes for anyone who wants to. And it looks like there's, like, source code and stuff available for it. They're not trying to keep it secret. So it's Telex.cc. And from there you can probably drill down and find their paper. I did. And I just perused it enough to get a gist of what they were doing. But it looks like it's the real deal.

I mean, it requires a lot, which is the downfall. It requires that essentially all the traffic that is leaving a censored environment after it gets out of its censoring would pass through this Telex, or whatever they're going to end up calling it, but this anti-censorship system because it is a man-in-the-middle technology. So if a given individual's traffic happened to go out through a different path, that is, bounced through standard routers and non-anti-censor-enabled ISPs, then it would just - it would go to its destination. They would get connected to where their packet obviously said they wanted to get connected to, which is not what they really want.

So with this comes a tremendous implementation burden, which is that, to be reliable, all the traffic leaving China, for example, would need to have, like, sweetheart ISPs who are all in on this, who are running this technology, examining all of the SSL connections that are being established and doing the cryptographic work of seeing whether this is one of

these tricky packets; and, if so, then doing redirections. So to me the downside is it's more of a capability than a solution because to get this thing to really work would require an awful lot of implementation side stuff.

TOM: But it's certainly progress. And I guess the censor could block an entire ISP if there was a Telex station on that and play a game of chicken and say, you get the Telex stations off your network, or we're not going to allow you back in. But that's sort of a scorched earth policy.

Steve: Well, it is. And they could also, the other thing is, the assumption is that the censor is allowing non-proxied SSL traffic. That is to say, the censor is allowing people within the environment to have traffic go out that they cannot see into. And I wonder about that because one of the first things that a censored environment would do would be to require you, if you want SSL connections, to install their certificate in your browser. In which case they're able to intercept and proxy your SSL, decrypt it, see what it is, and then reencrypt it.

Well, that decryption/reencryption and establishing a new SSL connection outbound would break the fancy protocol that this Telex technology does. So it would not - this technology would not survive having SSL essentially proxied by somebody who wanted to see what you were doing the way, for example, many corporations do that want to proxy for reasons of providing antivirus and more robust filtering. Well, that's the same thing that a sensor wants to do is robust filtering. So anyway, it's extremely cool technology. A bunch of people brought it to my attention, so I wanted to say, yeah, I looked at it; that's how it works. It's not clear to me that it's practical, but definitely very clever.

TOM: Well, and you need stuff like this to be discovered and explored to get to that practical technology that will do what this wants to do. This is a step along the way, for sure.

Steve: Right.

TOM: Let's move onto something that is possibly considered less cool by many. We've got a return of the undeletable cookie from KISSmetrics.

Steve: Yeah. Yup, another, again, lots of people brought it to my attention, wanted to understand what was going on. We've talked extensively, of course, because tracking is something that all of our listeners are concerned about, and monitoring it, and controlling it. The first tracking we know was from just standard browser cookies, so-called third-party cookies that were being hosted by websites that you then attracted these cookies through as a consequence of third-party resources being loaded by web pages.

TOM: Evercookies, sometimes people call them.

Steve: Yeah, well, an evercookie is a little bit different than that. That's a sort of a pseudocookie that's used by sort of - and that's still more sort of theoretical. There actually is something called an evercookie which uses lots of characteristics about your client in order to lock onto you. Then Flash began being used as sort of an off-the-map, off-the-radar approach. So sites would host Flash. Flash by default is configured to allow local storage on the user's machine. And by scripting Flash, it was possible to resurrect cookies that had been previously deleted. So users who were concerned about privacy began routinely, not only blocking third-party cookies, but also just deleting all of their first-party cookies because they realized that would still make them known to sites that they revisited. And in some cases, for whatever reason, they chose not to have that be

possible. Flash was being used then to recreate cookies that had been deleted. And so that was considered to be a problem.

It's these UC Berkeley privacy researchers who stumbled just recently on this KISSmetrics service. They're the same guys that brought the Flash cookies to our awareness back in 2009. So they published a paper just last Friday, July 29th, which was picked up by Wired and sounded the alarm that there was a new technology in place that defeated all known anti-tracking technology. It used HTML5 and something that they called "ETag Respawning."

TOM: At least it's standards compliant.

Steve: Yes. And unfortunately our browsers are now up to HTML5 and support ETags, and so unfortunately they support this KISSmetrics.

TOM: So what is ETag respawning? How does that work?

Steve: Well, okay. So the way assets are managed through the web is we'd like to be able to have our browser cache things so that we're not having to constantly get them over and over and over. So, for example, if you're on Amazon, all of that, the website's decoration, is in images which your browser needs to get once. But then as you go from Amazon page to Amazon page, even though the pages are different, all that window dressing is pretty much the same. So you'd like your browser to be able to just use what it already has in its cache.

There are a number of ways for a server to allow a browser to cache. For example, and this is normally done in headers that the user doesn't see that are so-called "metatags," or meta information, which is additional information. So, for example, there will be expiration dates on the information that the server says, essentially allowed the browser to know that it can, until a certain date and time, it can reuse this. The browser is able to send a query to the server saying, if this object has not been modified since a certain date, then just let me know that. Otherwise, give me the update. So browsers are also able to, like, send back sort of conditional queries saying this is what I've got that has this following date stamp on it. If you've got something newer, then I definitely want to know. Otherwise just let me know that it's not been modified. And so servers spend a lot of time these days sending back I think it's 301 Not Modified replies to browsers' conditional queries.

Well, there are problems with intermediate caches which can sometimes muck these things up. So the HTML standards folks said, we need something a little more robust. Let's essentially come up with a hash for these objects so that the server, when it sends an object to the client, it will also provide what's called an "ETag." "E" stands for "entity." So it's an entity tag which uniquely identifies this particular instance of any object with that name. So now the browser is able to say, I have an entity; I have an object with the following URL. And essentially I have its hash. It asks the server to verify that the entity hasn't changed, even by the same name.

So the bad news is this is - it's supposed to be an opaque token. Just a string of gibberish will be unique for a given instance of an object. That is to say, a hash. If you change the object - and the HTML spec doesn't say it has to be a hash. It doesn't actually say it has to be anything. It's just supposed to be a blob, an opaque token which is unique to this object. So you could use, for example, a good CRC algorithm, as long as you didn't have a problem with collisions of different objects having the same CRC, the same checksum. So it can be really anything; and it's not taken as a hash, it's just taken as something opaque.

Well, the guys at KISSmetrics, which was a San Francisco-based, Silicon Valley-style startup a few years ago, they said, we're going to come up with something that's even stronger for tracking than anyone has done before.

TOM: And that's what all these companies are looking for is the most robust tracking that they can come up with. They want to not be deleted. They don't want to have their ads blocked. They want to get around all of that stuff.

Steve: Yes. And so there's two aspects to this. To use the term "stovepiping," where you keep things within one stovepipe, there's the notion of identifying people who come back to you who are your people coming back to you in the future. That's one aspect. And users may or may not want to be identified by the same site when they return. Often it's valuable because, for example, you don't have to then reauthenticate and re-log in. You're able, for example, with eBay there is a checkbox, keep me logged in for 24 hours. And so there's a huge aspect of convenience for being identified both as you move around a site from one page to the next, but also if you return an hour later.

TOM: Yeah, and Google does this on a vast basis where, if you're logged in, you can stay logged in for two weeks. And when you move from Google Docs to Google+ to Gmail you're constantly authenticated.

Steve: Right. Then the second more onerous and worrisome tracking is of course cross-site tracking where - and this is the one that concerns most people, where because sites use, for example, common advertisers like DoubleClick that of course is now a Google property, because sites use common third parties, when a given individual goes to a different site that shares that third party, that provides cross-site connectivity. And this is precisely the behavior that these UC Berkeley privacy researchers found and verified. They found, first of all, this is not just, for example, Hulu, which was talked about in this article, and Spotify. There were a thousand of the top websites are all now using KISSmetrics.

TOM: Which is why Hulu and Spotify were so upset that they got mentioned.

Steve: They got singled out, exactly.

TOM: But there's 998 others out there, too.

Steve: Yeah, I mean, very popular sites because this technology is so powerful. What the Berkeley guys found, and this was what sent a chill down their spine, is the exact same ID, a long crypto-looking unique tag, existed on many different sites that were using KISSmetrics. Meaning that KISSmetrics was not assigning random tracking tokens, but was synchronizing tokens across multiple sites. Which meant that a given site using KISSmetrics could say to a different site using KISSmetrics, hey, here's the KISSmetrics token I have for such and such a user. Can you tell me anything about them? And oh, by the way, I'll share some information with you if you'll share some with me. So it really increased the concern about intersite collusion, which has been a problem in the past.

So the good news is this generated a big kerfuffle. KISSmetrics is now saying that they've backed off from their technology - and this is only in a few days. In a matter of five days they've now posted an update on their site saying that they're just going to use cookies for persistence. What it looks like they were doing is they were using custom JavaScript. It was a file called "i.js." And in this JavaScript they would embed the unique token for the user. Okay, so not removing cookies, not dealing with Flash, not doing

anything that users could do other than emptying their entire browser cache, that's what it took in order to shake this KISSmetrics technology use.

TOM: And that's because they were using HTML5 and ETag, because that's the place it's stored.

Steve: Yes. And what the privacy researchers at UC Berkeley realized was that this was the first time they had seen ETags being used in the wild for ID tracking, not just benignly for cache tracking. They were embedding IDs in ETags that were being provided by KISSmetrics. And so those things would live in your browser cache persistently. And then this i.js file would pull back the ETags and then regenerate information and allow much more pervasive tracking than we'd seen before.

I did want to let our users know, we've talked about Ghostery as a very cool add-on, a cross-browser add-on, I've got it both in Firefox and Chrome, which notifies you of which sites are providing third-party tracking when you visit pages. And I noted that KISSInsights and KISSMetrics are both appearing on various sites when I'm seeing my little Ghostery popup, so users can see where this is going on. And my takeaway from all this is that this is another example of state-of-the-art technology being used to track us; and that ultimately, much as I love technology, we all know I do, it's going to be legislative, a legislative solution that we end up, I think, generating and relying on. It'll be when there's the force of law behind something as simple as the do-not-track header in our browsers where sites are required by law to ask for permission to track us, rather than assuming by default that they're able to.

TOM: Even that won't get rid of all the instances, of course. But at least it wipes out a bunch of people like Hulu and Spotify who want to be law-abiding.

Steve: Well, yes. And, for example, it was when the use of Flash cookies came to light in 2009, immediately a class-action lawsuit was filed against the companies who were doing it. And at the time Flash was ignoring the private browsing options in browsers. So the Flash cookies were persisting into and out of private browser sessions. Adobe modified Flash in order to behave itself when Flash cookies were enabled and to give users the privacy that they were clearly explicitly asking for when they were using in-private browsing technology.

So I believe it's the case that there is enough concern, and this is clearly a subject of enough abuse, that it's going to be when browsers simply say I do not want to be tracked, that legitimate companies that are right now able to track us without our knowledge are going to have to do so with our permission or risk the consequences, which I think will be significant because clearly enough people are upset about this that good companies have to abide.

TOM: And in the meantime, KISSmetrics, we should reiterate, have said they've switched to cookies-only IDs. But who's to say they couldn't switch back later without telling someone, or some other company comes along and tries the same shenanigan.

Steve: Well, or a different shenanigan. And that's why I think, if you just step back from this, the technology is out of control. And so I really think the only solution is going to be simply making our preference not to be tracked known in a simple way and then having watchdogs like these UC Berkeley guys and so many others checking to make sure that companies are honoring it and just stomping on them with the force of law when companies don't. I mean, they're just - we can't stay ahead of it technologically.

TOM: Yeah, it's a constant race. Let's move on to an errata around the Blowfish bug.

Steve: Yeah. Many, many, many people commented. We talked last week about a bug in using Blowfish for hashing and how some confusion with the signed or unsigned-ness of characters could cause a dramatic weakening in the hash that resulted. Many people commented that ASCII is a seven-bit code, and so you'd always have the high bit off in ASCII code.

So first of all I wanted to acknowledge everyone who said that. You're absolutely right. However, the application for this code was not strictly ASCII input. So there was an instance or are instances where this could still bite you. But even so, people were saying, well, this isn't really as bad as we thought. Well, that wasn't the point of the podcast. The point of the podcast was to demonstrate an example where the programmer clearly made a mistake which the compiler was obscuring as a means of demonstrating how easy it is to make these kind of mistakes.

So I wanted to absolutely acknowledge that, if only seven-bit code was being used in this instance, then this particular bug would never manifest. But it turns out that input with high bits on is often used. But mostly this was not meant as, okay, how bad is this problem in the wild, but here's such a cool textbook classic case of how bugs like this do happen in real life.

TOM: Let's go on to, then, the Portable Sound Blaster project that you announced out on Google+. And you've got a Google Group for it, as well. What's going on with this?

Steve: Well, I don't know, Tom, if you're aware of the most popular and arguably famous Security Now! podcast of all time, which is called The Portable Dog Killer podcast.

TOM: Yes, yes. I have heard the tales.

Steve: So that was Podcast #248. And immediately after the podcast was aired, people began writing to me, asking for plans for their own version of this thing. And I will say again, because I'm - 40 years after I made this thing (I made it when I was 16, and I'm 56 today), the name makes me just cringe. I'm very self-conscious about having ever have called anything a Portable Dog Killer, especially when in fact I really believe it probably saved the dog's life that I used this device to train because...

TOM: It's a [indiscernible] trainer, maybe. No dogs, no animals have been harmed with the use of your application; correct?

Steve: Correct. Some seagulls had their flight paths altered. But that didn't hurt them, either. So anyway, a year later, or - yes, it was May 2010. And as people discover it, I'm getting constant mail. What I have come to realize is this problem of barking dogs is much worse than I ever really appreciated. Now, I created this to train a dog not to leap at a fence and knock people off the sidewalk and scare them to death, or near death.

TOM: Again, no humans killed in the use of this application.

Steve: But there are people who are unable to sell their homes, I've heard stories of people who cannot sell their homes because of the dog next door. My own parents put up with a dog that would bark all night for decades, asking the owner to please take care of their dog or something. But no, I don't know what's happening. Society's decaying. We have less sense of community now than we used to. I've heard stories about babies being awakened by dogs that people have no control over.

Anyway, it was finally, a few months ago, when I was enjoying or trying to enjoy an afternoon on a friend's patio, we were doing some backyard barbecue grilling, that this dog next door was just barking at nothing for hours. And it really was disturbing. Mark explained that this - and he had owned the house for a year at that point - that this had been going on for a year. And so I thought, okay. Maybe it is time to revisit this issue, to explore the possibility of training aberrant canines to stop barking.

So I took a break from the project I'm nearly finished with - I've been working on a very exciting new crypto thing that I will be telling our listeners about shortly. I took a break to look at updating the technology. And I've got that underway. Parts are on order. I created a group on Google called The Portable Sound Blaster. And so that's what I'm calling this thing. There's also a problem I have out on the patio at Starbucks early in the morning with, for some reason, on some mornings, an amazing collection of crows descend on one tree and squawk for about an hour. And I've seen other people, other Starbucks patrons screaming up at this tree, telling them to shut up.

TOM: Are you sure they're not ravens delivering messages from distant kingdoms?

Steve: I'm not sure of that.

TOM: Either way they're annoying, is what you're saying.

Steve: But they certainly are annoying. So I'm curious. I want to see how they would respond to something like what I created before, but updated four decades later using current technology. I don't know what the result will be. I also want to see about - I've got two friends, both named Mark, who are good friends, both with problems like this. I want to see if there is some solution and maybe offer - and I don't know yet because we don't know what the results will be, but maybe offer a solution to other people. So whatever it is that I do, I'm just going to make public.

TOM: And the idea is that this is a sound that the dogs hear and dissuades them from barking?

Steve: Maybe. I don't know. I mean, maybe they'll bark louder, in which case...

TOM: Hope not.

Steve: ...it would be counterproductive. So but I'm going to create a flexible device which is capable of sending a beam of sound between 3500 and 25 KHz, so through the audible range. The problem with birds is their hearing, their frequency response begins to drop off around 8500 and 9000 Hz, which is entirely audible to us. So anything that would convince the birds to leave the tree, we would be able to hear. It is not the case with dogs, whose of course hearing famously is technically supersonic. It's above the range of human hearing. So I don't know if they'll stop barking. I don't know if we could, like, if they bark we reward them or punish them with a blast of sound and then stop, and like they learn that, oh, maybe barking isn't what I'm supposed to be doing. I don't know. So there's lots of experiments to be had. I just wanted to get this thing started because I'll be working on it in the background. I'm going to immediately return to finish this crypto project that I'm very excited about, which I'll be talking to all of our listeners about before long. But I also wanted to announce that this is underway. Many people have asked about my use of the term "sound blaster," saying that Portable Sound Blaster sounds like something from Creative Labs.

TOM: Yeah, I'm going to take my audio card out and carry it around with me.

Steve: Exactly. So I thought I'd just take this opportunity to teach a little bit about trademarks. The problem with the trademark "Sound Blaster" is that it is extremely descriptive, which means it is a very weak trademark. So there is no trademark infringement if I have something that I call, whether it's a group on Google or a device, if I call it a Portable Sound Blaster, that's what it is. It's descriptive, and it is not infringing anyone's trademark. The problem...

TOM: And if you were making a sound card and trying to call it a Sound Blaster, you'd be in trouble because you would likely cause confusion. But this is a different kind of device, so you're not likely to cause that confusion.

Steve: Exactly. And, for example, trademarks like Xerox or Kleenex, they're incredible good and strong trademarks because they don't say anything about what they do. They're not descriptive. I mean, Kleenex, maybe, sort of. Maybe it's a concatenation of clean and some - and tissue, I mean, I don't know what Kleenex means, probably just the "kleen" part.

TOM: If they were tissues for daubing the neck area, then maybe they would be more descriptive, but they're not.

Steve: Right, correct. But they are - or, like, Exxon, fantastic trademark. I mean, it is owned, those things are owned by those companies. But something like "sound blaster," that was never a good trademark for Creative Labs to have because it is too descriptive. So for me to have something called a Portable Sound Blaster, I can call it that if I want to. As you said, it will not confuse people because it's not a sound card. But it's also - I'm just describing what this thing is. It is not a violation of anyone's trademark.

TOM: Although we'll see that put to the test with the Apple case, suing Amazon to stop them using the term "app store." And that's Amazon's defense is that's descriptive. Apple's defense is, no, "application store" would be descriptive, but "app store" is not.

Steve: You're right.

TOM: So that's an interesting application of that test.

Steve: There are gray areas.

TOM: So you've been getting into "Falling Skies"?

Steve: Well, okay. So here's prompted that. Yes, first of all, I've been watching it, and I like it.

TOM: I got sucked in, too. I didn't think I was going to like it. I totally like it.

Steve: Well, yes. And so it is the fact that there's a "Falling Skies" marathon on TNT Sunday that caused me to bring it to our listeners' attention. It's been a short season. It's, what, maybe 10 weeks? So TNT, Turner Network Television I guess is what TNT stands for...

TOM: Yup, unless it's Tech News Today. You've got to watch out for that trademark thing again.

Steve: There you go. So the TNT cable channel has been airing a sci-fi series called "Falling Skies." It stars, those of you who don't know, it stars Noah Wyle, who came to

everyone's attention for his never-ending stint on the show "ER," NBC's, I think, what, it ran 10 or more years?

TOM: More, I think, yeah.

Steve: Okay. A good friend of mine, Mark Thompson of AnalogX, hates the show. He only saw the first two episodes, and it just made him gag. But we've got very different tastes, Mark and I. It is true that when it tries to do sort of soap opera-y things, it stumbles because the actors are not fantastic. But it's good science fiction.

So I did a little poking around, and there is a "Falling Skies" page on Wikipedia for anyone who's curious. Tim Goodman of the Hollywood Reporter, reviewing it, wrote: "The entertainment value and suspense of 'Falling Skies' is paced just right. You get the sense that we'll get those answers eventually. And yet you want to devour the next episode immediately." Thomas Connor of the Chicago Sun-Times called it "...a trustworthy family drama but with aliens." He continued, "It's 'Jericho' meets 'V,' with the good from both and the bad discarded." He says, "It'll raise the summer TV bar significantly."

Ken Tucker from Entertainment Weekly gave the series a B+ and wrote, "A similar, gradually developed, but decisive conviction makes 'Falling Skies' an engaging, if derivative, chunk of dystopian sci-fi." He concluded, "'Falling Skies' rises above any one performance. It's the spectacle of humans versus aliens that draws you in." And finally, the Boston Herald, Mark A. Perigard, gave the series a B grade, writing, "Don't look now, but 'Falling Skies' could be a summer obsession." Well, okay. So I'm not giving this the Gibson sci-fi five-star award. It doesn't merit that. It's not as good as reading "Daemon" and "FreedomTM" or any of the Peter F. Hamilton stuff.

TOM: No, no. I agree. By the way, that's the 10th episode airing this Sunday.

Steve: Okay. So it was very short. It was last week that they said "only two remaining." I went, what? Oh, shoot. Because, I mean, I'm enjoying it. I think it got better. And there's enough good sci-fi in there. We're learning things about these aliens that have attacked Earth and are occupying the Earth. And so anyway, who knows if it will get renewed. That will be the test. Is it going to get renewed for a larger and larger chunk of time? We've seen really good sci-fi, like "Firefly," that didn't even have the episodes that had been made all aired. So networks don't always do smart things. I think it was worth watching.

So because the entire first nine episodes will be re-aired all day long, culminating in a two-hour season finale next Sunday, I did want to bring it to our listeners' attention. It's not too late. If this sounds interesting, start your TiVos and suck in the whole series and see what you think. If you subscribe to a box that's got TNT on it anyway, it's free, and I think it's been worth watching. I hope it gets a second season.

TOM: Madgician (sp) says it got renewed. I looked it up, and there's a quote from Michael Wright of TNT, the network, saying that it in fact will be renewed for a second season. So good news.

Steve: Yay.

TOM: The only criticism I have - I'm getting sucked in, too - is that I wonder why the aliens don't have aerial surveillance, why they can't just look down and see all the humans. Like it seems a little easy for them to hide.

Steve: Well, okay. Yes. And Mark Thompson's criticism is that. He says you're required to suspend too much disbelief. On the other hand, he watches fantasy anime. So it's like, okay, Mark. I can't watch that stuff.

TOM: I can suspend that much disbelief. It doesn't ruin it for me. But that was something that crossed my mind. So we've got a note from Jeremy Webb?

Steve: We do. Actually it was really well timed. It was this morning he sent an email to my sales folks, who forwarded it to me. He said, "Another satisfied SpinRite customer." And this one's sort of interesting. He said, "Dear Steve, I've always been my parents' tech support guy. When I joined the U.S. Air Force, they stationed me pretty far away from home. Thankfully, I've always been able to VNC into their computers and get things straightened out for them.

"This week I was presented with a rather unique challenge. My mother called to tell me that their computer was throwing a bunch of disk errors in Windows. Fixing this problem was particularly difficult for two reasons. First, VNC wouldn't help them if their hard disk crashed suddenly. Second, I'm currently deployed to Afghanistan, where getting a good enough connection to VNC into their computer can be difficult. I knew that SpinRite might be able to fix the disk errors, but would I be able to walk my parents, who aren't the most tech-savvy people, through it over the phone?

"Believe it or not, I was able to get a good enough connection to VNC into their computer and make for them a SpinRite bootable image. I was able to instruct my computer over the phone how to boot into SpinRite and start the repair process. She called me the next day and told me that SpinRite had fixed 12 errors and that their computer was now back to normal. I cannot tell you how much we appreciate your product. It really saved the day. Jeremy."

TOM: That's a feat, VNC'ing from there. That's great. Let's move on to our listener feedback, the password-insecure 1-2-3 episode. Don't use the number of this listener feedback as your password.

Steve: Do not. Oh, there's one thing, though, I forgot to mention. Anyone who wants to get to, at any time in the future, get to that Google Group for the Portable Sound Blaster, I added a link to it in GRC's main menu. So at GRC.com, the last menu item says "Other." And that's just sort of my random catchall. So under "Other" you will see "Portable Sound Blaster." And that'll take you to the Google Group. So for anyone who wants to know how to find it easily.

TOM: There's one question answered before you even asked it. Now you know how to find it. So let's start off with Steve - oh, I'm sorry, you're Steve. He's writing to you. His name is Alon. He's part of the Social Media Team at GoDaddy.com in Scottsdale, Arizona. And he wrote in to mention what he believed may have been a factual error in Security Now! 308, where we discussed choosing to drop GoDaddy.com as a provider of SSL certificates. Obviously he wishes that weren't the case, but he understands you have to have the freedom to choose the provider that best fits you.

He's referring to the pricing mentioned for our SSL certs. He says, during the show it was stated that our certs cost thousands of dollars. I wanted to clarify that. Not only do GoDaddy.com EV certs not cost thousands of dollars, I believe they're actually the least expensive EV certs on the market today. A single extended validation certificate or premium SSL currently costs only $99.99, less if purchased in multiple years. And furthermore, this pricing isn't new. It's been this way for some time, and at no time has this type of cert cost $2,000 as suggested on the show. So he goes on, he says, I fully

respect your decision to publicly applaud the services you love. But he wanted to point out that pricing difference.

Steve: Well, I did respond to him because he got a bunch of his facts wrong. He assumed that I was currently using his certs, and that it was GoDaddy I was leaving and intending to go over to DigiCert. It is VeriSign that I have often and almost constantly talked about as having amazingly expensive certs, and that I intend to go over to DigiCert because, thanks to the Certificate Patrol add-on on Firefox, I saw that Facebook was using DigiCert, and a number of other very high-profile sites, so my feeling was, if they can, I can, too. So that gave me the courage to leave VeriSign, where I have been from the beginning, just because VeriSign was like one of the very early founders of all of this technology.

However, I did want to remind our listeners that neither Leo nor I will ever use GoDaddy because my problem with GoDaddy is they attempted to charge my credit card for an expired cert without my permission. They asked me several times in email. There was no way for me to say no. There was no way for me to tell them I did not want to have my certificate renewed. And then they sent me a note that my credit card had failed their attempt to charge it.

The good news was, through some miracle I had used a PayPal one-time-use credit card number during the time that PayPal offered that service. And so GoDaddy was unable to zap me again on that card. But when I realized they had tried with no authorization from me, I told Leo, and we both swore them off. So they may be cheap, but they're just off my radar. I'm going to be switching to DigiCert here toward the end of the year when the first of my certificates...

TOM: So the confusion, the price you were talking about was VeriSign's price.

Steve: Yes.

TOM: But the issue was not price-related with GoDaddy.

Steve: Correct. I wasn't leaving GoDaddy. I had left GoDaddy. And I will be leaving VeriSign to go over to DigiCert.

TOM: All right. Question #2 from Tom Zerucha in Metropolitan Detroit, Michigan wonders about disabling tracking on tablets. He says, on an iPad 2 or Android tablet, how do I stop third-party cookies or tracking sites or the other stuff if I can't install Ghostery, NoScript, Cookie Monster or the rest? Which you can't, on those browsers. Is there a hosts file that I can point the tracking sites to 127.0.0.1, or something else?

Steve: Well, I can't speak to Android. Maybe you can, Tom. I do know that on the iPad, under the settings, the main settings app, for the default browser, which is Safari, you can tell it that you only want to accept cookies from sites you visit, meaning not third-party cookies. I sort of think that's the default, if I remember. I checked, and mine is set that way, only allow from sites I visit. But I think maybe Apple is unique in this industry for setting all their browsers that way by default, which in fact I'm sure of that because I've looked at third-party cookie stats, and Safari's stats are way down almost at zero compared to every other browser. So that's one of the nice things that Apple has always been doing for us is blocking that. The problem is you're on a very restricted platform where we just don't have things like the hosts file or the ability to intercept DNS and redirect things. And for one thing, we also don't have Flash, so we don't have to worry about Flash cookies over on iPad 2. But, Tom, are you an Android person?

TOM: I'm not, but my wife is. And I know some of my best friends are Android users. If you push the menu button on your phone, and then you get more options, you go to the "more options," you can select a "settings" function. And in the settings function there's an "accept cookies" option that you can uncheck, and then it will not accept cookies anymore.

Steve: So that's a default...

TOM: And that's in the default browser, anyway.

Steve: Right. And I was going to say also, I also looked, in wondering whether, for example, iTunes offered a privacy-enhanced browser. I know, for example, that the LastPass folks have an iPad browser that binds the LastPass functionality in to make it easier to log in. You don't need to use scriptlets and things, which you otherwise normally have to use in Safari. And I couldn't find anything. But I would imagine at some point maybe someone will do a privacy-enhanced browser that offers explicit do-not-track me additions, which could be very nice.

TOM: @msx in the chatroom points out that both iPad and Android have an ETC hosts file. So that makes sense that, if you can get in there and edit that, you might be able to do something there.

Steve: So I would bet that Android then you do have - because you do have a much more open and less controlled environment than on the iPad, you could go in and do that, yeah.

TOM: If you jailbreak your iPad, you'd be able to get at that hosts file, most likely.

Steve: If there is one. I mean, we don't know that there is one.

TOM: Yeah. There's a directory. I don't know if there's a file. That's a good point. Brian Lawson in Lebanon, Indiana wonders about PIE and online backup. He says, I've been wondering something about the combination of PIE and online backup services like Carbonite. I have a 1TB drive and have created a 500GB TrueCrypt file on that drive for the data I want to keep under tight lockdown. I created a TrueCrypt-encrypted file instead of a partition because services like Carbonite will back up a whole internal drive; but if it were segmented into partitions, I believe I could only pick one or the other drive letter. This way I can point it at the whole drive and back up both the encrypted and non-encrypted data because it is on the same drive and drive letter.

Now, since Carbonite works on differential backups after the initial run, would it have to reupload the entire 500GB TrueCrypt file every time it changes? I know you can do things in TrueCrypt like disable the timestamp update. But the contents of the file will change with each update, and therefore the pseudorandom noise that is the file on the drive will look different to Carbonite. It will recognize it as a file that has been updated and grab the new version. I think I know the answer to the question, and I'm not sure I like it. But I thought I'd check with the experts before dropping dollars on this approach.

Steve: Okay. So he did a good, clever thing. He created a big file, and so he is hosting a TrueCrypt container inside that file, which is on his drive. Carbonite is smart in the same way that - I'm trying to think of the other service. I think it was Dropbox we were just talking about where it will take a large file and segment it in much smaller pieces, on the order of, like, 16MB, and generate hashes for individual pieces. They do this to save themselves bandwidth and to address exactly this concern, which is, if you had a huge

file backed up, but only part of it were to change, Carbonite would only want to zero in on that part that changed and save themselves all of the bandwidth, and you all the bandwidth, of re-uploading the entire thing.

The way TrueCrypt functions, it is encrypting blocks that are sector size and nothing larger. So even though the whole TrueCrypt file, which is containing an entire drive, looks like pseudorandom noise, changes that are local within the file system will be reflected in local pseudorandom noise changes in the file. So the good news is, Brian, it's exactly what you want. You will only see the same kind of uploads to reflect changes in the overall TrueCrypt file that you would see being made in the directory system. So there'll be some changes in the metadata that manage the files, and then the files themselves that come and go. But once you get that big 500GB blob up once, then only little changes to it will have to be updated. So that's a really great solution.

TOM: And I should point out, and a bunch of people in the chatroom are saying this, too, all your internal drives can be backed up. So if you wanted to do an entire encrypted drive, it could still be backed up separately from other internal drives, even if you partition. I've done that before with Carbonite.

Steve: Do you know whether Carbonite's view of the drive is pre-TrueCrypt or post-TrueCrypt?

TOM: Oh, now, that is a good question. I don't know. When I said I've done it before, I've backed up multiple internal drive partitions. I have not had them encrypted at the time when I did that.

Steve: Right. My guess is that TrueCrypt would have the same view of the drive that the OS does.

TOM: Yeah, no, that makes sense. So what he's doing is a clever way to make sure that it stays encrypted.

Steve: Yup.

TOM: All right. Joel Davis in Albuquerque, the ABQ, New Mexico has some info on NTP. He says: I just finished listening to Security Now! 308, and I wanted to give you updated information on NTP and Internet time in general. I work for an electric utility, and we use NTP to sync clocks with extreme accuracy to do fault location on transmission lines. We're able to get around 10 "us" sync, as compared to GPS clocks, out of NTP. We are currently in the process of upgrading to IEEE1588 as a protocol, which will do about 1 "us" accuracy over packet-switched networks. What is that "us"?

Steve: Microseconds.

TOM: That's microseconds. Oh, he's using the letter "u," but he means - okay, now I get it. Will do about 1 microsecond accuracy over packet-switched networks. Also I wanted to give you some better information on the NERC decision about power system frequency. This is a nonissue. In the mid-'90s NERC switched from an IFE, an Integral Frequency Error, to an AFE, Absolute Frequency Error standard. This means that instead of having to bring the integral of daily frequency error to zero for the last 15 years or so, utilities have only had to cross nominal frequency once an hour. Now NERC is removing that requirement. It is still in the interest of the utility to maintain frequency close to nominal, and the regional entities, WECC and ERCOT and the others, still have frequency deviation standards for their individual areas. Sorry so long, but I hope it helps. Joel.

Steve: So just to remind our listeners, we talked back on Episode 308 about the report that power-generating entities around the U.S. were going to be allowed to let their frequencies wander and drift more than had been the case historically. And Leo and I hailing back from the, well, the '50s and '60s, remember that original clocks depended upon the exact power line frequency. They actually were essentially counting cycles electromechanically with their own motors synchronized to the 60Hz power.

And the idea was that, during times of extreme power load, the frequency of our AC line would tend to droop a little bit. It would drop below 60 as the generators in the power plants, whether it be coal fired or hydroelectric, they were put under a greater load, and the actually spinning generators slowed down that were generating the AC waveform. And then the idea was that at night the engineers would have been counting cycles all day, and they would run fast in order to make up for the fact that they were literally cycles behind due to the load during the peak power delivery of the day.

And so what he's saying is that actually the world has been cut loose since the '90s; that they are being further cut loose. But the presumption is that by this point we're no longer in a cycle-counting mode, that our clocks have built-in crystals that they're depending upon for accuracy. And so the absolute number of cycles we've had of AC just isn't that big a deal any longer. So I'm glad to know.

TOM: Yeah, good stuff. Thank you, Joel. Question #5 comes from Geir in Norway about last week's Blowfish bug, again. He says: Browsing the web for more info on this Blowfish bug you talked about in last week's Security Now! episode, I came across this page at Schneier.com/blowfish-bug.txt. It includes an email from 1996 which seems to point out exactly this same bug, already known back then. It even suggests both solutions you described, either casting the variable type or just declaring it as an unsigned int. Thought you'd like to know.

Steve: So sure enough, I went to this page. I don't know when Bruce put this up on his site. But it's Schneier.com/blowfish-bug.txt, and it shows a clear, I don't know if it's email or newsgroup posting. It's something that's got all kinds of time and date stamp headers all over it, where there is a discussion and showing the code doing exactly this wrong thing back then. So it was known to people, and somehow it just never got fixed.

TOM: History repeats itself.

Steve: Yeah, boy.

TOM: Joe in Pequot Lakes, Minnesota says: In Episode 299 and 301 of Security Now!, as well as a Q&A episode, you alluded to sources of random numbers. While there are USB, PCI, and PCIe cards that can be purchased to allow your computer to generate true random numbers, there are some sites on the Internet where you can request truly random data sequences generated from some really creative sources. And he lists Random.org, which I've actually used before, generated from atmospheric noise; and Fourmilab.ch/hotbits, generated from radioactive decay of Caesium-137. These sources can be queried via SSL over the Internet for bits of pure random data for programs or via a browser. Silicon Graphics once provided random numbers via Lavarand, taking pictures of patterns produced by a lava lamp. Just some fun sources of randomness.

Steve: So I just thought that was cool. I wanted to share those sources with our listeners. Using contemporary browser technology, AJAX-style queries, it's possible for script in a browser to establish a connection to any of these sites and get some randomness for its own use. So...

TOM: I used to - we would do a drawing on a show I did for a prize. And I used to use Random.org because what I liked about it is you can put in a parameter. You can say, give me a number between this and that, and it will give you a random number. And it'll give it in whole integers and everything like you said. It's really good.

Christopher Hopper in Brisbane, Queensland, Australia has an annoying bank. He says: I'm a recent subscriber to Security Now! and so have been made aware of your recent revelations on password security, entropy, and the best way to construct a password that is hard to guess using brute-force password-cracking techniques. Thanks for thinking on these matters and sharing your own epiphany with us. I liked that episode, too, where Steve came up with his new password advice. I've been implementing that.

Getting back to the email, he says: I've taken your advice proffered on the Password Haystacks page to improve my already fairly strong master password and make it even stronger. I'm already a LastPass user, so my master password is very important and must be strong and easy to remember. To achieve this I have applied a "leet" method of spelling to an easy-to-remember code. To make it stronger, I'm now padding it out another four spaces using a special character.

I am concerned, though, with my banks. I have two online banking accounts with two separate banks, and in both cases I am not allowed to use special characters in my password. One of the banks has a character limit of six to eight characters, which is just lunacy. I don't see a reason for it. And I know, thanks to your podcast with Leo, just what it might mean on the back end. I'm a web application developer myself, so I'm aware of how passwords are stored in back-end databases using salted hashes. I don't see why, if they're doing it right, they need to restrict either the type of characters or the number. Is there a security article from a trusted, known source, written in plain, easy to understand language, that I can point the banks to, to explain to them why they shouldn't put restrictions on password length or composition?

Steve: Okay. So, first, to answer Christopher's question, I'm unaware of somewhere I could point him to. But I thought that I would put it out there for our listeners. Maybe someone knows of something clear and clean and simple. I mean, obviously I could write something, but that's not my job. Maybe something exists. The problem is that I'm not convinced banks care. Certainly they're getting complaints from their customers. And in answering the question of why banks do this, the only thing we've been able to come up with is that it's purely from a customer service standpoint. They want to provide a password that they're able to read over the phone or somehow communicate with. And customers are entirely capable of generating passwords that are just gibberish that no one, no bank personnel could handle. Certainly we know we have all the technology required to easily handle really strong authentication. It just - it's difficult to understand why banks are doing it, other than inertia. I imagine that 10 years from now this won't be a problem. But it certainly is now. I don't know whether sending a note to a bank would have any effect on them. I sadly think it probably would not.

TOM: You're absolutely right about that, customer service, I agree. I think it's because they don't want to deal with people who can't remember their password, so they keep them short. Because the other thing they do which annoys me is, even if they do allow long passwords with case sensitive and special characters, they then say "and you have to have a security recovery question." And that security recovery question has to be in plain English. So you've just undermined the entire password at that point.

Steve: Exactly, yup.

TOM: All right. Gary W. in Detroit, Michigan mentioned that Steve didn't get a chance to

explain the red division error in SpinRite. He was just listening to the last podcast, and you guys switched into the main topic before you had a chance to explain the "why" portion of the SpinRite's testimonial question of why it displayed the red division error one time, but succeeded the next time. What's the answer to that?

Steve: Okay. What's going on is that SpinRite is still using some functions of the BIOS. And SpinRite is probably one of the few utilities around which is still robustly exercising the BIOS. Most operating systems now use the BIOS just enough to get themselves loaded, reading a simple range of sectors into RAM, and then as soon as they can they switch into protected mode. The BIOS is real mode on the Intel architecture, not protected mode. It won't work in protected mode. But all operating systems run in so-called protected mode, where they get advantage of all the extra features the chip has to offer. So they were in a hurry to get into that mode in order to finish booting themselves.

What happens is that, as a consequence over time, BIOSes are beginning to be a little buggy. We all know that, if code is not being checked and is not being exercised, it has a tendency not to be correct. And so in some edge cases, SpinRite will, while it's doing data recovery, will cause the drive to respond in a way that upsets the BIOS, which is essentially an intermediary between SpinRite and the drive. And so that we look at those division errors, and we see that it's in the BIOS, so there's nothing we can do about it.

The good news is SpinRite, as did happen in the case of last week's testimonial, SpinRite can recover the sector anyway. The problem tends not to persist. And in this case, in the case that we shared last week, the user ran SpinRite again and didn't have the problem recur. So it is an interaction between the BIOS and a drive which is doing something that is upsetting the BIOS. And the top of my list for what I will do when I next move SpinRite forward is to disconnect it completely from the BIOS. It will become more like an operating system that just gets itself going and then no longer uses the BIOS for anything once it's running. So that problem, which isn't a big problem for our users, it will become a zero problem.

TOM: The BIOS is a cruel and fickle mistress.

Steve: And it's lonely. No one really uses it anymore.

TOM: Pete Costello in Cheesequake, New Jersey - which apparently means 'Upland" or "Upland Village" in the local Native American language. I had to look it up to find out why something would be called Cheesequake. Might be pronounced like Ches-a-cake or something. Anyway, he has extensive experience with compiler evolution and sign-ness. He writes: Your "Anatomy of a Security Mistake" brought back nervous ticks from my days compiling embedded systems code for many large firmware-based telephony systems. I think there's a possible explanation for how such bugs appear in old code: changes in compiler behavior. That is, compilers change.

I would run my own compiler tests whenever the compiler vendor or open source would release a new version. The test consisted of "if" statements with unsigned versus signed comparisons of every permutation of C-type char, unsigned char, short, int, long, et cetera. What I had found was compilers were not identical, even between revisions. The "if" path taken would vary between many compiler versions, the fix being to explicitly cast as unsigned wherever there was a test. Consequently, it is possible to test the source with one compiler, fine. But if compiled later with a different compiler, the behavior would be different. Perhaps that explains how old code can suddenly be found behaving unexpectedly. And of course this is a huge concern for security-related code where the cost of a mistake can be much greater than the program not working as designed.

The confusion started during the formation of the ANSI C standard definition of the language. Early C compilers from Bell Labs "preserved the sign," while Borland and the ANSI committee thought that was not intuitive, wanting to sell C compilers to the masses, and chose to "preserve the value." The result was two incompatible compiler types in the programming environs yielding two different path-taking results. The ANSI committee even created several compiler options allowing a transition to the new C compiler behavior for those customers with existing code bases .

But even into the 1990s I still found compilers to exhibit different behaviors. My company went to lengths of buying me the Plum Hall C Compiler Validation Suite to ensure that the large code base we burned into the hardware was not going to be replaced because of variations caused by a compiler upgrade. I'm thinking this can explain how code can be developed and even tested, but later with a different compiler exhibit buggy behavior.

Steve: I thought that was great news. Being an assembly coder myself, I haven't been plagued by these problems. But a number of listeners, and I wanted to acknowledge the others who sent similar notes, basically said very much the same thing. This signed versus unsigned comparisons apparently has been a bane of C programmers' lives for decades. And I loved Pete explaining a little bit of the history of how we got into this trouble, that Borland and the ANSI committee thought, well, rather than preserving the sign, we're going to preserve the value, which is just nutty. It's like, well, okay. Gosh.

TOM: I'm sure they had their reasons.

Steve: The implications for security, we covered it in detail last week. So many subtle errors can be caused by that kind of boundary condition.

TOM: Question #10 comes from Ken Giurlando in Lancaster, Pennsylvania, writing about "Zero Day," "Daemon," and "FreedomTM": I've been listening since day one, he writes, and I've been happily using SpinRite for many years. Having just read all these books you recommended and loved them, it seems like the focus should be preventing rootkits and changes to the BIOS. Short of that we should have an easy way to install a clean OS. If, for example, in "Zero Day" they had been able to reload the OS in total, as is done with the iPhone, they could have been back up quickly. I'm sure it is more complex than that. And of course reloading the OS would have to be done from an external safe OS. But what is stopping us from doing this now?

Steve: I would say two things. First of all, the size and complexity of the OS. I mean, I'm sitting here in front of an installation of Windows XP. I would love nothing more than to somehow easily flush it out and start over. But as all Windows users know, the OS and our apps get so inextricably entwined that you have to quit the entire system and start over. It's just not possible to, I mean, well, okay. I have a friend who is crazy, and he tried to maintain application-ness separately from the OS. But he spent more time doing that than he did getting work done.

And the second thing is that the original iPhone didn't have these problems because it was completely closed and had no apps. We'll all remember that the first iPhone, this whole app store thing, and user-provided apps, that came afterwards. It arguably wasn't part of Apple's original plan, this amazing mother lode that they stumbled into; and creating the apps store wasn't part of their original idea, which is why it took them so long to come up with application programming kits for people who wanted to add apps to phones.

My point is that any closed platform can be vastly more secure than any open platform.

And the more open it is, to the degree that it's incremental - for example, the first iPhone had zero openness; now the iPhone has relatively good openness. But, for example, the Android platform is wildly open. And so we see exactly a correlation between security problems and openness that track. So, unfortunately, no one is willing to have a closed platform. They're not willing to trade that for security. Everyone says, oh, we wish we had security. But then someone says, okay, but what if you can't run your app on it? Oh, well, then I don't want it at all.

TOM: Yeah, there's a mobile security firm called Lookout that says Android users are two and a half times more likely to encounter malware than six months ago. Isn't the argument that open can be more secure is that people can find the bugs faster and squash them, not that it's inherently more secure?

Steve: Yeah. Well, the idea, of course, the idea of openness is that you'll have many more eyes on it, that in a closed environment only the authors know what they're doing, and they're inherently biased toward wanting to believe that it's correct. Or in some cases, I mean, we've seen evidence of known problems going unfixed because those people just don't care, or you might say they don't have the right priorities or the right time. But in an open environment you'll always have somebody who says, wait a minute, I've got a free night. I'll fix this right now, and then merge the fix into the rest of the code stream.

TOM: And we should point out, in the books "Daemon" and "FreedomTM," the daemon would have corrupted your OS already. You wouldn't have had a secure version of it most likely.

Steve: Exactly.

TOM: Finally, Question #11. Quentin Roberts in St. John's, Newfoundland, sends "a message from Matthew Sobol." Thank you, Steve, for recommending "Daemon" and "FreedomTM." I just finished reading "FreedomTM," and it was amazing. Do you have any other books similar to this series? I'd love to read more like it as my mind is just blown by all the possibilities and how technically accurate it all is. I've been a long-time listener of Security Now!, and I have been listening to older episodes as I am working, and it has really refreshed some old material I've forgotten over the years. I enjoy listening every week. Keep up the great work. Got any other book recommendations for Quentin here?

Steve: Well, okay. I put this question in, assuming that you would be Leo. Because, as our listeners know, when I discovered "Daemon" and "FreedomTM," Leo was like, well, duh.

TOM: Well, I'm the one who told him about "Daemon."

Steve: I was really late to the party, apparently. And it was after I finished reading "Zero Day" and people were tweeting me saying, Steve, if you liked "Zero Day" by Mark Russinovich, you're going to love "Daemon." And it's like, I am? So I said that to Leo, he's like, duh. We've all read that, Gibson. Where have you been? So I thought maybe Leo would have any other ideas. To me, these have just been, I mean, I'm a sci-fi fanatic. This is a little more real; and, as he said, "it really could happen" and "technically accurate." I think those are the things that Quentin said he specifically liked about "Daemon" and "FreedomTM." And that's sort of not the kind of stuff I normally read, although certainly I am loving these.

TOM: Now, I know Daniel Suarez is getting close to finishing the third book in that

installment. So...

Steve: Whoa, whoa, whoa, whoa, whoa. Is it the third of this...

TOM: It's a new book. I should be careful in saying it is part of the installment because I actually don't know how much in that universe it takes place. But I know it exists in that same arena, let's put it...

Steve: Oh, and Leo said he's going to have him on.

TOM: Yeah, we're going to try to get him on, definitely. I don't know that there's a whole lot else besides "Zero Day" by Mark Russinovich that's really like this. I mean, this is sort of a new genre budding out. I mean, there's "Neuromancer," and there's traditional cyberpunk.

Steve: All the old classics.

TOM: Yeah. And Cory Doctorow, one of my favorite Cory Doctorow stories is "When Sysadmins Ruled the Earth," about sort of this disaster happening, and the sysadmins are the only people that can communicate with each other for various things. So there's some stuff out there. But really not a lot. I've heard about "Fatal System Error" by Joseph Menn, but I've never read it. That would be the only one that I could throw out there.

Steve: Okay, cool.

TOM: Well, that wraps it up. Thank you, Steve, for allowing me to fill in for Leo. I know he'll be back next week, barring getting selected for the jury, which I don't...

Steve: You did a great job, as always, Tom.

TOM: Well, thank you.

Steve: You powered through these questions, so that was great.

TOM: You can find Steve Gibson all over the Internet. But the place you've got to look is GRC.com. That's where you find SpinRite, that's where you find ShieldsUP!, that's where you find all the stuff going on. In fact, the new, not dog herder, but the Sound Blaster is - you can find the link there, as well. Anything else we should mention before we go?

Steve: I think we got it. Our listeners know about SpinRite, that pays all my bills. And I really appreciate the support of those who purchase it even when they don't need it. And next week, unless some strange disaster befalls us, I plan to do another installment of our "How the Internet Works," probably talking about ICMP and UDP protocols. So we'll have a little bit of an interesting deep-tech propellerhead episode once again.

TOM: You may or may not realize you've run into UDP protocols out there. So that's a good one to pay attention to. All right, thanks everybody.

Steve: We're talking over one right now, as a matter of fact.

TOM: That's right, exactly. All right. Thanks, everybody, for watching. You can find us at TWiT.tv/sn. Leo will be back next week, we hope. See you then.

Copyright (c) 2006 by Steve Gibson and Leo Laporte. SOME RIGHTS RESERVED

This work is licensed for the good of the Internet Community under the Creative Commons License v2.5. See the following Web page for details: http://creativecommons.org/licenses/by-nc-sa/2.5/