SECURITY NOW!

Transcript of Episode #204

Listener Feedback #70

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-204.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-204-lq.mp3

INTRO: Netcasts you love, from people you trust. This is TWiT.

Leo Laporte: Bandwidth for Security Now! is provided by AOL Radio at AOL.com/podcasting.

This is Security Now! with Steve Gibson, Episode 204 for July 9, 2009: Listener Feedback #70. This show is brought to you by listeners like you and your contributions. We couldn't do it without you. Thanks so much.

It's time for Security Now!, the show that covers all things security/privacy related, with Mr. Steve Gibson, the man of the hour, the guy who discovered spyware, coined the term "spyware," and just disappeared as I pressed a button on the screen. Hi, Steve. Welcome to the show. Your show.

Steve Gibson: Hey, Leo. Great to be with you. As our listeners are hearing this, you are somewhere over the water in Asia.

Leo: [Singing] Somewhere over the rainbow. Let's see, this is July 9th. So, yeah, I will actually be on the boat sailing from China to Korea right now, exactly right.

Steve: And if you heeded the advice that we discussed two weeks ago, you're carrying an ATM card that is not backed by too much cash.

Leo: That's right.

Steve: Because you don't really control the environment over there to the degree - in fact, maybe no one controls the environment over there to the degree that we enjoy here in the U.S. So I hope that your cash is safe.

Leo: I don't know, you know, I mean, I think in the old days travel was a great adventure, and all sorts of horrible things could happen to you. And now with the Internet and television it's just become, you know, and air flights, it's just become pretty mundane. So maybe a little adventure.

Steve: It's very much more like you never even left home because, after all, there is a Starbucks on every corner, no matter what country you're in.

Leo: I was warning Henry that, oh, you know, the food in China, you know, may be very difficult for you. We might have to go through a week of suffering in China. And then somebody told me, oh, no, there's McDonald's everywhere. It's like, oh. They said the best Kentucky Fried Chicken you ever had is in Beijing. It's like, oh. Oh.

Steve: But as a consequence of the fact that we are recording this two weeks ahead, or three weeks ahead, actually, of when it's going to be aired, I don't have any security news. However, what we're going to do to make up for this is the Q&A in two weeks is just going to - is going to be our mega three weeks' worth of security news catch-up.

Leo: Everything that happened while Leo was in China.

Steve: Everything that happened while Leo was floating around Asia somewhere, we'll be talking about.

Leo: But to Steve's credit, he doesn't want to miss a single episode. He's never missed an episode. We are at 204 episodes in 204 weeks. So we decided to double up a couple of shows and make sure that we get everything recorded so that you miss nothing.

Steve: Well, I hear so many thank yous as I read through the mailbag - by the way, that's at GRC.com/feedback. So many people appreciate the reliability of knowing that they're going to have a Security Now! podcast absolutely every week, come hell or high water, that I don't want to disappoint them. They're going to have to wait a little bit for news. But they'll get the news that happened only a couple of weeks late. So we're going to do a special mega news catch-up podcast for the next Q&A two weeks from now.

I did have, when I was going through the mailbag for this particular story, a really nice sort of short SpinRite story that I wanted to share with our listeners from a woman named Louise who said that her - the subject was "GRC is my new hero." She says, "Hi, Steve and Leo. I have a SpinRite story for you. Recently someone using my computer managed to get it infected with a trojan."

Leo: Ugh, don't you hate that? Don't you hate it when that happens?

Steve: Yup. "Being a long-time Security Now! listener, I decided to reformat the drive" - good girl - "and reinstall Windows." Because after it's infected, that's what she means, is after - she knew that after it was infected she could never trust it again. So she said, "I backed everything up to my external drive. This drive locked up while I was attempting to move my data back onto my newly reinstalled computer. I shut everything down, checked all my connections, and tried again. Imagine my horror when I discovered all of my backed up data had disappeared. The drive still worked, but my folder was mysteriously empty. I prayed for a miracle." So that means she was able to see the drive, but the folder where she had put everything had no contents.

Leo: So that's kind of a good sign.

Steve: Well, it's better than if the drive was completely, completely not spinning anymore, or hosed in a bad way. I mean, yes, it's a hopeful sign I would say.

Leo: Hopeful, there's the word I was looking for.

Steve: So what she prayed for was what she needed. She says, "I prayed for a miracle and booted up my copy of SpinRite. (I had originally purchased it three years ago after losing two hard drives in grad school.) It took 27 hours, but I'm happy to say my data is all back and currently being copied back onto my laptop. Thank you so much for SpinRite. It is definitely worth every penny." So, neat story. Thank you, Louise.

Leo: Wow. Very interesting. Well done. Are you ready, my friend, for some questions and answers?

Steve: We've got - this one, okay, I put this at the top. Because we're recording double shows this week, we have a little bit more limited time than usual. Normally this would have been the absolute mind-blowing story of the week, which we would have put at the end. But I didn't want to risk missing this one because this is just too wonderful.

Leo: Nothing wrong with starting with a bang. And this is a bang. Trevor in Wall, New Jersey takes the cake with his Story of the Week. Subject: Windows in Places It Shouldn't Be - Supermarket Edition. Hey, Steve and Leo. I've been a listener since Episode 20, and I've listened to all of it in reruns. I certainly agree with Steve's attitude towards Windows, a consumer operating system, being used in mission-critical places. I recently had a horrifying experience at my local A&P supermarket here in New Jersey. I walked up to a self-checkout aisle and began to scan my items. Out of nowhere the system freezes, beeps, and begins to reboot. Lo and behold, a friendly Windows 2000 boot screen appears. I cringe. I cringe. As a long-time Security Now! listener, I watched in horrified fascination as the machine automatically logged in a user named "Terminal6." There was an icon on the desktop labeled "Shopping Panel" that probably was the interface for the checkout application.

So I decided to do a little investigating. I opened Internet Explorer 6, and it brought me to MSN.com. Hmm, guess they never changed the default web page. At this point an annoyed A&P employee walked over and took control of the machine. He launched the checkout application and walked away. There was no antivirus software on this machine. There was no software firewall. And this machine had direct access to the Internet, running probably unpatched Windows 2000 Professional. Good lord.

Steve: I know.

Leo: I am never putting my credit card into one of these self-checkout machines ever again. Thanks for all the great work on Security Now! as well as Leo's other - I'm so glad that this person listens and had the sense to try that.

Steve: Isn't that wonderful?

Leo: Unbelievable.

Steve: I mean, I just - I get the hugest kick out of, okay, we're surfing the Internet from the self-checkout machine in the supermarket.

Leo: It's connected to the 'Net, unprotected.

Steve: Launch IE6 and up comes...

Leo: Never patched, obviously, yeah, if it's running IE6.

Steve: Ugh.

Leo: By the way, he says: I convinced my dad to purchase SpinRite last year, and it saved him from losing his entire novel. We have been SpinRiting every two months ever since. There you go. That must make you feel good.

Steve: It does. That's just great.

Leo: Somebody's using your stuff to save their life.

Steve: Yeah. Anyway, Trevor, thank you, thank you for the story. I just - I get the biggest kick out of, you know, he's scanning his items, and it's Windows, so of course it was going to freeze sooner or later, and it chose to do it right then. Then it restarts itself and doesn't launch into the app, just gives him a desktop where he sees IE6, which he probably double-taps on the touchscreen and, whoop, up comes IE6 and brings up the news on the MSN.com site. I just love it.

Leo: I guess he could have run Windows Update, but that would've probably broken everything. You broke our computer.

Steve: Oh, goodness. And as he says, I'm not putting my credit card in one of these suckers again.

Leo: No, I don't blame him. Nor will I.

Steve: Wow.

Leo: I don't like those automatic checkouts anyway. But now I really - of course you can't, you know, they're probably using the same system for the checkers, I mean, it's probably exactly the same system.

Steve: Yeah, and Terminal6.

Leo: Terminal6. Question 2, Doug Zuckerman, Bethesda, Maryland, asks about finding Q&A questions: Steve, just a suggestion for the future, I think it'd be a great addition to the website, but wouldn't it be nice if on the main Security Now! page in the topic title for each Q&A episode of Security Now! it showed the specific questions you answered in that episode. I find that not only do I listen to your show for new information, I also use it for reference. And it's difficult to find the right Q&A shows to go back to when the questions aren't listed. Anyway, of course your site has tons of great resources as it is. I'm not trying to suggest you guys don't do enough already. I just think it'd be a really good enhancement if you ever feel like it.

Steve: Well...

Leo: I think we might do that ourselves in our show notes; don't we?

Steve: Well, the problem is that the questions are often hard to summarize. I mean, what would I write for that question that Doug just asked?

Leo: Right. Yeah, you're right, it's a good point.

Steve: And the good news is that we have the transcripts for everything, and those are indexed by Google and other search engines. I had an experience just a couple days ago of where, I don't remember, I was at Starbucks, of course, and I was looking something up, and the second link that came up in Google search results for the keywords that I was searching was one of our own podcasts. And it was #120. Oh, I put in "TrueCrypt" and "BootIt NG" because I was talking to a buddy of mine who had dropped by, and he was wondering whether you could use TrueCrypt and BootIt NG at the same time. And I said, gee, that's an interesting question, because I could see that there could be a boot sector track zero collision, much as there was initially with TrueCrypt and some of the Adobe copy protection stuff, the Macromedia stuff. So I put into Google "TrueCrypt BootIt NG." And in Security Now! Episode 120 we were talking about that. And so it was the second link that Google brought up. I thought, hey, now, this is exactly why we're doing transcripts.

Leo: That's really the point, yup.

Steve: Aside from the fact that people really do like to read along or read them instead of listen to them in some cases. And so we've got Elaine doing a great job of that. So, Doug, the only thing I could suggest is that on the Security Now! page at GRC, as you noted, SecurityNow.htm at GRC, there is a search box on the upper right. And it will search only Security Now! podcasts if you configure it that way. If you ask for that, it will limit itself to Security Now!. And so put in some keywords, and you can probably find the podcast. I think that's much more practical than trying to summarize the questions in each of the Q&As, which they're just - it'd be really hard to summarize them.

Leo: Well, but I also will point you to our wiki.twit.tv. We do show notes for all the shows there, and we have very detailed show notes, including the questions there for each episode. And on FriendFeed, as we're doing the show we have a FriendFeed conversation for every show that we do on the network. If you go to FriendFeed.com/twit-conversations - I don't think you have to sign up to see these, FriendFeed.com/twit-conversations - you can go back through all the shows. And as we ask the questions of Steve, I post the entire text of the question in there. So there are a number of places you can go. These will also be Googled. But there are a number of places you could go to find that information. But our wiki.twit.tv has very complete show notes, including timecode. And I don't know who's doing that, but I thank that person for doing all that effort because that's a heck of a lot of work to put in there. So thank you for all your hard work on that. Actually, if I look at the history I might be able to see. We have, it's nice, Steve, we have listeners who use the wiki, which is a media wiki, it's the same as Wikipedia, to really enhance the content of the website.

Steve: That's great.

Leo: And we do that for the same exact reason, I mean, the problem with a podcast, it isn't searchable. So having text related to the content so that you can search on Google is so important, I think, as time goes by. So thank you, Doug, for that question. And thank you to AKalair Mind Tricks for putting in all that information in our show notes.

Brad Banko in Cuyahoga Falls, Ohio - isn't that where Bailey Savings & Loan was? I don't know - asks about Assembly language. I listen to your podcast from time to time. Chess Griffin of the Linux Reality podcast pointed me to it when Vista opened my eyes to the need for an alternative to Microsoft software. By the way, I use

Fedora Linux quite a lot, really love it. 30 seconds to boot, 30 seconds to desktop, 30 seconds to shut down on a 10-year-old machine. The only thing that I still use Windows for are my scanner and TurboTax. My question is, you mentioned you develop your software in Assembly language. Doesn't that mean your work is tied to a particular, presumably Intel process architecture? Or has Assembly language programming advanced to the point that you can write architecture-independent code? My background's mainly science, with computing as a hobby and a tool. Thanks, enjoy your show.

Steve: Well, he is exactly right. What Assembly language is, is the actual instruction set of the machine. So I've been programming Intel Assembly code ever since the IBM PC happened. Before that it was the 6502 chip on the Apple II machine and also the Atari and the Commodore machines. So the 6502 was on multiple platforms, so I was able to use a single programming language among those manufacturers who were all using that chip. In this case, of course, the Intel architecture is dominant. It, for better or worse, won out over the PowerPC because people liked the idea that the clock rate was faster on the Intel chip, even though you could argue that the PowerPC, being a very nice RISC design or heavily RISC-influenced architecture, was doing more per cycle than the Intel was. Still, people wanted Intel chips for whatever reason.

So it is the case that I'm definitely writing platform-tied code. But I'm writing it for almost all the machines out there. I mean, I guess you have PDAs, for example, which are using ARM chips, and I couldn't use Intel Assembly language for writing to that platform. On the other hand, there is, when you're an Assembly language programmer, you're dealing with binary registers and motions of data among registers. And you're at a low level such that moving everything you know to a different platform isn't difficult. I mean, if someone said, Steve, we need you to write something for the ARM processor, which I've never programmed before, I would just sit down with the instruction reference and go, okay, how does the ARM add? What's the instruction for subtract? What's the instruction for load and move?

So I can easily map everything I know, all the experience that I have on one low-level architecture onto another. It's just a matter of knowing what the instructions are. Just as I went back, you know, turned the clock back and was writing code for PDP-8. It's like, oh, what a quaint little machine. But, you know, all the rules still apply.

Leo: Well, you have even more dependencies than just the code because you use - don't you use interrupts on the hardware?

Steve: Oh, yeah, I mean, you're right, it's not just the instruction set. It's the whole architecture.

Leo: I mean, you could in theory, I mean, Mac runs Intel. But you're writing not just to the processor, you're writing to the BIOS. You're writing to...

Steve: Correct.

Leo: So it's not portable beyond a PC.

Steve: All the hardware resources of that machine; right.

Leo: Which answers the question people often ask me, why is there no Mac version?

Steve: Right.

Leo: So, yeah, I mean, you get advantages with hardware dependence. But you also - that's the disadvantage, it's not portable. That's why I think most of the world's moved to higher level languages for most of the code.

Steve: Way most. Way, way most of the world.

Leo: Brian Taylor, Walnut, California, wants to know Steve's secret. What is your secret? I have a quick comment and a question. I've been listening to your wonderful netcast since Episode 1, lately have gotten way behind. I'm still in January. We've just made him a lot more behind. I just finished listening to an episode where you mentioned the PDP-8. I immediately headed over to SpareTimeGizmos.com to find out he sold out of the front panels this spring and has no intention of making more. Oh, well, my fault for not staying up to date.

As for my question, how do you do it? How do you wrap your mind around all the complexities of what you do? Is it an inborn ability, or is there something else in the mix? After listening to one of your episodes my brain feels like it ran headlong into a brick wall. Help me, Steve, I want to be smarter, I need to be smarter. I always think of it in this way: I'm just smart enough to know how dumb I really am. I'm with you on that one, Brian. I'm right there with you.

Steve: Well, okay. I've been wanting to say something along these lines for a while, and that is that you and I, Leo, have been around for a long time. So I'm drawing from a large base of stuff that I learned a long time ago. And I also have extensive knowledge about a relatively narrow category of things. But this format of the podcast does allow me to seem to have more facts at my grasp than I do. I spend, not a day, but many hours preparing for the podcast. And so, for example, I don't have RFC numbers memorized, or I don't have MD5 hashes of things memorized. I focus on the topic of the podcast for many hours beforehand, brushing up and bringing myself up to speed, getting all revved up and ready to go so that during an hour, an hour and a half, I'm at the top of my game on whatever topic we're discussing. And a couple weeks later it's all a blur.

Leo: Oh, thank you for admitting that.

Steve: It is.

Leo: It's all a blur to me a couple of minutes later. But at least - well, you know, as often is the case when somebody's really working hard, it makes it look easy. And you work really hard to make it look easy.

Steve: Well, yeah. I mean, I love the topic. I love the technology. I understand the fundamentals of this stuff. But nobody could know everything that's going on in every corner of the industry. And I certainly don't. I mean, I've got a day job. And I'm focused on one area. And I would argue where I'm focused, it's hard to fool me. But I can't talk about just that all the time, so we're talking about much more. And so I'll pick a topic. I'll say, ooh, this is something neat that I think would interest our listeners. And I then focus myself on that with the advantage of a lot of background that helps me digest it quickly. But the excruciating minutiae and details, again, it's like somebody will - sometimes I run across a question about something from a few weeks ago, and I go, oh, boy, we were talking about that. What is the answer to that one? You know?

So anyway, I just wanted to say that, in response to Brian's completely understandable question about wow, you know, how is there so much here, it's like, okay, Brian, I'm cheating. It's here today; it's gone tomorrow.

Leo: I forget it all very quickly, too. And I make more mistakes than you do. Mat Ludlam in Weybridge, London wants a bit of help with cookie programming. He says: Steve, I love the show. I've been a long-time listener since the very first episode. And I'd like your opinion on the subject of storing data in cookies. I want to store a cookie - he's obviously writing web applications here. I want to store a cookie on a user's machine that allows them to automatically log into my system. I want this cookie to have a lifespan of about seven days, and I don't trust the users. I don't have a problem with users seeing the information, I just don't want it changed.

My thoughts are to create a cookie which has the information I care about - login, password, et cetera), an expires date, and a hash total. I would take each of the pieces of data and put them through a hash along with some salt which is one of your secure passwords. This would be the hash total that I store on the browser cookie with other data. The salt would never leave the server, my server, so it's a secret. Is this system resistant to the data being changed?

On a related subject, I don't want to store the password in a cookie in cleartext, so what would you advise? If I store the password hashed with the userID, is that enough? Or do I need to add some more secret salt? Finally, how can I stop the user from copying a cookie from one machine to another? If I try to add the IP address of their machine, then I probably have the address of their company's NAT router, so it does not get tied to a single machine, rather an organization. And finally, am I doing this the right way, or is there a far better way of doing it? Thanks in advance for your help and advice. A little cookie programming advice.

Steve: Yeah, and an interesting question. I would say there, surprisingly, sort of is a right and wrong way to approach this problem. The biggest question that I would have for Mat is does he have the ability to store stuff at his end, on his server. We know that he can have keys and salt. But the question is, can he store information? Does he have the ability, does he have access to some sort of a database, even a simple database that he could write himself, sort of a flat file database. The point is, you either store the data in the cookie, the data itself in the cookie, or you just store a pointer to data that you're maintaining on the back end, on the server side.

So it's certainly not necessary to go through any of these gyrations if you've got the ability, if you've got state, if you can store state of some sort on the server end. In which case you would just take a counter, you'd take, like, a 128-bit counter, and you'd encrypt it with your own secret symmetric key. That would turn the counter - which starts at zero and runs up to, you know, all one bits - it would turn it into a completely pseudorandom-looking token. And so that the unencrypted counter would just be a record number in your own little private database. You encrypt it, and that's what you store in the cookie.

So the cookie itself has no data. It just has this, as we used the word two weeks ago, a "nonce," a one-time-use little blurb which is completely random-looking. The user could change it if they wanted. But when it came back to you and was decrypted it would be nonsense because you would know what the valid range of values was, zero to some limit. And if they made any change to it, it would send it off, when that was decrypted, it would completely turn into nonsense. So there's a simple solution, but it does require that you're storing no data in a cookie, you're just - essentially you're storing a pointer, an encrypted pointer, to a record in your own database back on the server. And of course then you've got all kinds of flexibility - username, password, expiration date, you can put as much data there as you want to and manage that.

If for some reason you cannot, you don't have the ability to have a database, to have any state stored at the server side, then you have no choice other than to store that data in the user's browser in a cookie that you give them. And there again the solution is simple, and that is you simply use your own secret encryption, your own secret key to take the data that you've got and concatenate it into a record and encrypt it. You'll want to hash it so that you generate a signature, so that it's essentially digitally signed using the techniques we've talked about before. Use one key for a keyed hash in order to create a digital signature, and then encrypt that with a different key.

That's going to be a blob of data which you can safely put on the user's browser. It's just, I mean, it's nonsense for them. Again, they can't change it. It means nothing to them. There's nothing that they can do to gain any leverage. Anything they do to, like, mess with it will just break it. And then every time they have a browser transaction, apparently you want to use that for allowing them back into your website for a period of seven days, well, that cookie would be sent back to you. You'd decrypt it with your symmetric key. You verify the hash of it using a different key. That tells you that nothing has happened to change it. And with every single transaction you get all the data back that you want to store - username, password, expiration date. You're able to then look at those things, decide what you want to do with them, and permit or not the user to proceed.

So either in the first case you're storing the data at your end and just giving them a little token for it, a pointer to it; or, in the second case, you give them all the data to store in their cookie, and they give it back to you every time. So either way will work, and it's just a function of what you're most comfortable with and what makes sense based on the capabilities you've got at the server end.

The last question you asked was, is there a way to prevent them from moving it from machine to machine? And I cannot see any way to do that without involving scripting. If you've got scripting involved, and the login facility can have JavaScript running, then you could certainly incorporate something into the cookie which binds to something unique about their machine.

Leo: Can you see their MAC address? Can you...

Steve: You could use the MAC address. And, for example, if this was on Windows, there are various GUIDs, G-U-I-Ds, for Windows. There are various things that are pretty much guaranteed to be unique from one machine to another. So if you incorporated that into the cookie, and then when they attempted to re-log in, you would again use some script to verify that. Then I could see how you could lock it to a given machine. But you'd have to have something active on the client side. Otherwise a browser is a browser is a browser. And if you move the cookie to a different machine, then it would be able - it would look the same from your viewpoint, from the server end, a client making that request. You really wouldn't see any difference.

And there's metadata that browsers send along with their queries that might be different from one machine to another, like what version of the browser, what version of Adobe and PDF reader and other things sort of tack themselves onto the queries. So you might see some difference. But if you really need to prevent it from being moved, you have to do something with scripting locally on the machine. And I think that solves the problem.

Leo: Steve, let's talk tethering.

Steve: Okay.

Leo: John in Indiana wonders about the security of tethering his cell phone to his laptop: Thank you both for your time. Leo, I can thank you for turning me on to the podcasts, so many so I can't keep up. All of TWiT's shows help me stay on top of the tech news, and that lets me help people that come to me asking for advice. So thank you for your time and energy. You're welcome. Security Now! is still my favorite. Yay.

I'd like to hear your thoughts, Steve, on the security of connecting to the Internet on a laptop via tethering from a cell phone. This is something they've added to, you know, the Palm Pre does it, I think. They've added it to the iPhone once AT&T says okay. I guess you're not behind a router, as you usually suggest. Can I assume that I'm at risk, that not having a router makes me less secure on a cell phone access? And does the same thing happen with these cell phone access cards, the EVDO cards? What about those?

Steve: Well, it's an interesting question. The only reservation I have about a cell phone tether as opposed to a cell phone access card is that we have seen instances of cell phones themselves being compromised.

Leo: Right.

Steve: That is, you know, it's a computer. It's running an operating system. It's got a browser. It's got instant messaging. It's got all those things that tend to have problems. And we mentioned two weeks ago that when the iPhone was updated to v3.0, more than 40-some security problems were patched. Many of them were like, can take over your phone remotely and execute remote code, et cetera. We know that many phones also have Bluetooth, and Bluetooth is another vector of real security concern because people often leave them in discoverable mode. And there are many well-known exploits where you can take over somebody's phone using their discoverable Bluetooth pairing.

And so the problem is, when you combine that with its use as an Internet modem and tether it to your laptop, you're essentially hooking two computers together, one which tends to be exposed to a lot of threats. So as opposed to a cell phone access card, which is just a dumb cellular modem, and itself cannot be compromised. So from a security standpoint I would be much more comfortable using a cell phone access card than tethering my phone to the computer.

But aside from that, the cell phone service itself, that is, you know, the technology of cellular communications, is secure, but not nearly as secure as is state of the art. It is definitely the case that all of the different cell phone technologies can be cracked and have been cracked. They go to a great extent to obscure the traffic. And so I would say it obscuration rather than security because, for example, some of the cell phone technology used cleverly designed shift registers with prime number repetitions to create pseudorandom data which is mixed in, then XORed with the stream of data leaving the phone.

Well, it means that it's good security, but it's far from the type of security we talk about on this show as being absolutely uncrackable. And it has been cracked. I mean, all of the cell phone technologies have been cracked. There's no question that NSA people and even much less skilled hackers are able to get into that conversation. So you should consider that your use of a cell phone communication is not much more secure than an open WiFi hotspot. I mean, it is more secure than that because a hotspot is just sniffable with anything. But and so it takes much more technology to crack a cellular connection. But it is definitely crackable in a short, relatively short length of time.

So you should think in terms of using a VPN, making sure you're using an SSL, you know, HTTPS connection when you do things, the sorts of things that you would do if you were in an open WiFi hotspot in order to have the kind of security that you probably want to have.

Leo: I have an interesting hybrid product I've been using. It's called a MiFi. And it takes an EVDO connection via Verizon. And it has a hotspot WiFi access point built into it. So it is a router.

Steve: That's the worst of all worlds.

Leo: [Laughing] Yeah, you just made me think that. Well, the router, I turned on WPA. So at least I have that. But so they'd have to crack the EVDO to make that...

Steve: Yeah. And again, it's not easy. But every one of the cell phone standards has been cracked. They've got - the problem is...

Leo: Wow, I didn't know that.

Steve: ...they were designed a long time ago. They were designed back when we had much weaker hardware. So in the same way that the original WiFi with its WEP was designed, it's like, okay, this is good enough to secure, well, the hardware grew in capacity, but a standard had been established. Similarly, these cellular standards, I mean, the hardware to implement them is worldwide, spread out in towers and in pockets all over the globe. You can't change it now.

And the bad news is, it was regarded as strong back then, but also it was a clear compromise because the hardware, the cellular handset hardware was so low power in terms of computing power that they couldn't do much more than they did. So they said, well, this is as good as we can get now. The problem is, technology has made huge leaps since then. The standard hasn't changed. So you really, I mean, if security is a concern, you want to provide your own wrapper, your own security wrapper around your traffic so that, if anybody - I mean, and I doubt this is happening. And remember that, you know, the NSA doesn't have to crack it in the air. They just have to go visit the base station at the other end, and it's been cracked for them.

Leo: They have access to everything, presumably.

Steve: There's much easier ways to do this, yes.

Leo: If you've got the clout. Brandon in Portland, Oregon wonders if size really does matter: Dear Steve and Leo, I have a question about hard drive data integrity and how it declines as hard drive storage and density goes up. I've heard you mention in the past that, as hard drive density goes up, the margin for error goes down, which in turn requires more error correcting due to the density at which the data's stored. I thought I heard, and correct me if I'm wrong, that your comfort zone for hard drives is around 500GB and lower. Anything above that density has too large a chance for error for Steve.

My question is this. Is a terabyte drive really any less robust than a drive with much lower density? Isn't a terabyte drive usually four 250GB drive platters in one drive enclosure? In this case, wouldn't a terabyte drive be just as reliable as a 250GB drive, seeing as how the data is being stored on essentially four 250GB drives? I'd love to know the verdict and why especially, since you make everything that requires a propeller hat so simple to understand - oh, especially you, since you do that. Love the podcast, listening since day one and looking forward to it every week. Next Friday is my paycheck at a real job, so I'm looking forward to buying my first copy of SpinRite. And I've been a TWiT supporter and donator since day one. Thank you. Thanks for both of your hard work. Thank you, Brandon. So actually I don't know if that's the case. What you're talking about is areal density, the amount of bits squoze onto a single platter; right?

Steve: Squoze, yeah.

Leo: Or a single cubic centimeter, I guess.

Steve: See how Elaine spells "squoze." I have to confess that I think this is largely me just sort of being an old curmudgeon.

Leo: Curmudgeon, yes, that's the word I would use.

Steve: Yeah. Brandon is correct that you'll notice that drives jump in capacity by sort of fixed numbers, which is the number, is the amount of bytes that go on a platter. So you'll have, like, the 40GB and the 80GB and the 160 and the 320. And it's like, is it any surprise that these things are jumping by that amount? That is, literally, they put another platter in, and it doubled it. Or another one, and now it went up by two thirds of that size and so forth. So it is the case that they develop a technology that fits a certain density on a surface. And in some cases they'll leave one surface unused because that's what their marketing department tells them to do, even though it doesn't make any sense logically.

Now, is four 250GB platters the same reliability if they're put in a single drive, as if they were in four drives? You could argue that if you had them in four separate drives you could run RAID on them. So you'd lose the equivalent of one drive, yet that would give you dramatically more reliability because then, if any one drive died, you've still got a hundred percent of your data. Whereas the more data you put in a single enclosure, it's sort of like the problem of the airliner that crashes. It's like, well, boy, you know, we lost a lot of people when that happened. Whereas you also say, well, more people die on the road every day in little onesie accidents that don't draw nearly as much attention.

So, I mean, the reliability question is an interesting one. I would have to say that contemporary drives are extremely reliable, but you never want to be in a position where losing one really creates a problem for you because they die. And so, for example, as I have said before, there's no system I have where I have mission-critical data where I don't have a RAID and some sort of on-the-fly, in-the-background backup happening. You really do want to make sure that a catastrophic loss of a drive is not something unrecoverable because drives die.

Leo: Yeah.

Steve: And I'm glad that Brandon's going to be buying a copy of SpinRite, so he'll be prepared when one does.

Leo: Drives die. That's the bad news. So are you buying terabyte drives now?

Steve: I've had no occasion to because Assembly language doesn't take up that much space, Leo. No, I'm kidding.

Leo: We buy six of them at a time, you know? And we record everything on terabyte drives. And I'm sure we'll go to a bigger size at some point. But right now the sweet spot, I mean, a Caviar Green Western Digital drive is about 85 bucks from New Egg.

Steve: Wow, wow, wow.

Leo: For a terabyte. And these are really good drives. They run a little slower, but the areal density is so high they don't feel slower. But the RPMs, I think, are 5400. So I think that - of course, this isn't mission-critical, either; right? I mean, these are just recordings of the shows and stuff.

Steve: Yeah, it's just what we do all day long.

Leo: But we have other - we have other recordings. These are more like archives. I don't, you know, I actually don't do anything with them right now. But we'll have them. I don't know if somebody will want them someday. Look back, the good old days...

Steve: Good to keep them.

Leo: ...back before aliens took over the world and eliminated all security flaws. Scott Teriano in Port Pirie - not sure how you say that, Port Prairie? - South Australia says Steve's non-VPN idea is partly original: Hi, Steve, Leo. Your idea - I'm sorry, I won't do this - your idea of a non-VPN is indeed a fantastic idea, and much better than the traditional VPN. However, something very similar has unfortunately been done in the past. In fact, I'm currently running a similar setup myself. I was shown how to do this by my cousin, who is 16 years old, who had implemented everything you described except for the encryption and configuration part of what you described using OpenBSD's PF, which is their Packet Filter program. I have my implementation and IPtables under Linux since both my client and my router run Linux, and I have the Netfilter kernel module loaded.

The two things your solution cleverly applies, which mine and my cousin's do not, is encryption, which is obviously important if you have sensitive information, and the easy configuration. Writing PF or IPtables rules by hand is definitely not for the faint of heart. I know, I've done it. Of course, if you're using Windows, neither IPtables nor PF are available, so I guess it's unique for those folks. So is what he's talking about similar to what you were talking about?

Steve: Okay. This is an interesting - this question caught my attention because he describes - he says something very similar has "unfortunately" been done in the past. As if the fact that I'm doing something similar to what was done before is unfortunate.

Leo: Like it wasn't completely original or something.

Steve: Yes, and that's exactly it. There was somebody in our newsgroups, when he heard about my idea with the Star Trek Bunnons of reversing...

Leo: You didn't invent that, you know.

Steve: He was disappointed because he realized that I'm so old that I probably did it before he did.

Leo: Oh.

Steve: And I wrote back, and I said, wait a minute, I said, it's not about who did it first. It's that you had all the fun of being confronted with a problem and coming up with a solution. No, Leo, I mean, this is something I feel so fundamentally, it's a weird thing, I don't know if I'm strange or there's something that I'm not understanding. But like I independently invented stateless TCP connections, which I later found were called SYN cookies, and they've been done. And it's like, I didn't feel at all diminished because I independently invented something really cool. So what if I wasn't first? I came up with it. And like this guy posting in the newsgroup who said, hey, you know, gee, I really thought that that was mine, unique. Well, it was yours. You know, if you did it by yourself, you had a problem, you scratched your head, and you came up with a solution, that's the joy. I mean, that's the benefit, the fun of being out there, being creative, and coming up with solutions.

So I don't feel in any way that it's unfortunate, for example, that the idea of port shifting was done before or isn't unique with me. I didn't really think it probably was. It was a problem, and I found a solution. And so I just sort of want to address, because this did come up in the context before, and it has before, and no doubt as I'm doing this CryptoLink VPN there are things that are going to be absolutely new, but I'm also going to be treading well-trodden territory, doing things other people have done before. I want to solve the problem myself. I don't want to take somebody else's code. I don't want to take somebody else's idea. I want to scratch my head and say, okay, great, people have done it before. I'm going to challenge whether theirs is the best way it can possibly be done. People have done router traversal. Great. I haven't done it. Maybe I'll come up with something better. Maybe not. I don't know.

But, I mean, the joy is in doing it. The reason I'm wanting to tackle this is there are problems I haven't faced before. And so I just - I don't get that psychology, that sense that, I mean, I understand from an intellectual property standpoint and patents and who came first and who came second. That's got to be honored, and I respect that. But the idea of independently solving a problem, that's where I think the value is. That's where you learn something. I'm going to learn a lot writing CryptoLink. I can't wait. Even if a lot of it's been done before, I don't care.

Leo: There is that funny thing going around that you have to be first. I think the Internet really encourages that. You've got to be the first to do something or say something or post something. And you're right, you get the pleasure out of doing it, regardless of whether you're first or not.

Steve: Yeah, I just don't get that it diminishes what I do in any way that it isn't first. Hey, it's mine. So it's been done before. Okay, that was theirs. And the guy who came up with reversing English twice, good for him. I mean, yeah, I did it. There was no Internet to publish it, and I don't care. It was just fun for me to do it.

Leo: I had fun doing it, yeah.

Steve: Yeah, I'm glad he came up with the same solution. Bravo.

Leo: All right, here's one from Tom Shuman in Minneapolis. We were talking on an episode some time back about teaching teenagers computing. Hi, Steve. Thanks to you and Leo for the program. I'm a long-time listener and SpinRite user since version 2 - wow - on my spacious 20MB Seagate ST-251. Oh, I had a few of those.

Steve: Yeah.

Leo: I remember when 20MB was a lot.

Steve: It's all you needed, Leo.

Leo: I used SpinRite to change the sector interleave - oh, I remember doing that - on the drive, so my XT clone in "turbo mode" would approach the full 7.44MHz of the processor. That's right, there was a default interleave. But if you wanted to tweak it, you could.

Steve: And remember the XT clones? There was like a compatibility mode where they would run at 4.77, which was the original speed. But, oh, you could juice them up to 8MHz. And then it was, like, really fast.

Leo: I had a machine like that, and it never was stable.

Steve: No.

Leo: It could not handle 7.44. And, you know, we just had all sorts of stability issues. 7MHz. My phone does that when it's just thinking, when it's just waiting for me, I mean, it's ridiculous. Leo has expressed interest in teaching a programming class. Let me say first I have no vested interest in what I am about to suggest. Just another idea he might consider. ISECOM, that's the Institute for Security and Open Methodologies, has a program called "Hacker Highschool." I like that.

Steve: Yeah.

Leo: It teaches security awareness for teens. Oh, we're definitely adding that to the curriculum.

Steve: Yes.

Leo: While I don't like the name of the program - hacking is not what it used to be -

the program itself is awesome and would be terrific if taught by someone with a background in programming and a background in the history of the industry, a sense about how we got where we are today. Perhaps the gang at Tech Guy Labs would find the program interesting fodder for a series of videos. Hmm. I can't imagine anyone else that could do it better. Check it out at hackerhighschool.org. Just a thought. Keep up the great work. Thank you, Tom. What a great suggestion.

Steve: Yeah. I want to say that I looked at it, and it is really nice. And, I mean, I immediately, you know, high school kids, they would love the idea. I mean, hacking seems glamorous and immediately hooks them just much more so than bit twiddling would. And anyway, the site is really nice, hackerhighschool.org. And so I wanted not only for you, Leo, to run across it, but to also aim our listeners at it because they may well have their own teenagers or know some that might get a kick out of this, too.

Leo: So it teaches you how to be secure online, basically; is that it?

Steve: Yeah. It's, well, it's just - it's like, yes, security awareness, like what's going on, how does the stuff work. And, I mean, I can't think of anything more important for teenage kids to get a grip on as they're moving out into the world. And computers and laptops and cell phones and the Internet is tomorrow's reality. Only it's today's reality. But certainly it's going to be with them for their entire lives.

Leo: Neat.

Steve: And apparently not much more secure than it is today.

Leo: Yeah. Brad Beyenhof in San Diego, California wants a little more detail on CryptoLink. He says: Hi, Steve. In Episode 200 you read a letter of mine asking about the NAT traversal capabilities of your non-VPN CryptoLink solution. You mentioned that advanced users will be able to configure port forwarding, but a third party will be available for those who won't or can't do so. Are you going to be providing this third-party service? If so, what will be the terms of that service? I'm of course referring to the possibility of subscription fees and so forth. If you're not going to provide the third-party service, will you be relying on others to develop their own CryptoLink connection services? Or will there be a solution for advanced users to offer themselves as that third party using a coordinating service similar to those used by BitTorrent, TOR, and SETI@home? What's the plan?

Steve: I have no idea.

Leo: Good answer.

Steve: I mean, again, I didn't really intend this to feed off of my prior rant about first invention. But really, I know where my heart is. My heart is that buying CryptoLink gives you access to everything it is for life, forever. No subscription fees, no recurring anything, nothing. I mean, that's really what differentiates it from service-oriented offerings which are very different from the tool. People in the newsgroups have asked, hey, will there be a per-system fee of some sort? The answer is no. That is, an individual who owns it can use it for their own purposes as much as they want, on as many systems as they want, wherever they have a problem that CryptoLink can solve.

I'm hoping to have the most sophisticated and capable NAT traversal system ever created. I mean, that's one of the reasons I want to do this. I just don't want to say, oh, I have it. I want to be able to say mine nails anyone else's NAT traversal that's ever done it before. And so all kinds of technology will come out of this. I know that, for example, we will have a web-based NAT router characterization service, much like we have now ShieldsUP! testing your security, the DNS spoofability system that we'll be talking about before long, and Perfect Passwords and other stuff. There will also be a NAT router characterizer where users will be able to find out exactly how their NATs work because I'm going to have to know that in order to design the technology to traverse them. So there'll be things that fall out from this research which is really going to be fun and interesting and useful.

I imagine that GRC will have to provide that rendezvous service because I think, my plan is, that I'm going to be doing things that have never been done before to make the NAT traversal more robust than anybody else's. At the same time, I don't like the idea, as I've said, of anyone depending on us. If GRC was DDoSed, and we were gone, I wouldn't want for people not to have an alternative. So my intention is that the protocols will be open, that it would be possible for someone else to create such a third-party service. There are publicly available NAT traversal facilities. Maybe you could downgrade to one of those if you didn't want to use GRC, or couldn't.

But it's also the case that you don't need to use that at all. If you were able to map a port through your router, then you would be able to connect to anybody, and anybody would be able to connect to you. So a lot of this is going to be a journey. I know it'll be interesting for everyone who wants to participate in the newsgroups while I'm churning out this technology.

Leo: Great. But there is going to be some cost if you do a third-party server. It wouldn't be unreasonable to charge for that. I don't think anybody would think any the less of you. That's how you do the NAT traversal is you have a third party that they both connect to?

Steve: I would never do that. I mean, my plan is for the ongoing sales of CryptoLink to provide for the service. I mean, the fact is, Leo, bandwidth costs nothing these days. I mean, I'm running servers. And I would not be, ever, I would never be proxying the data. That is, for example, the way NAT traversal works in the GotoMyPC case, for example, is they are transiting the data. Both endpoints connect out to them, and so they're transiting the data. It's trivial to do that, but it does require that they then be a carrier capable of handling all the bandwidth. My approach is never to be involved in the data transit. I just want to introduce the two endpoints to each other and so that somebody looking at the traffic, somebody with a packet capturer can see that GRC or any other rendezvous service is not involved in that traffic. Thus it follows my TNO, my Trust No One model, of saying we don't want to be involved, so this is provably secure.

Leo: Moving along, a note from Paul Scott in Las Vegas, Nevada, a quick note about

Apple's AirPort Express. He says: I love the show, have been listening from day one. I fell a little behind, and while catching up heard a comment from a listener that he uses AirPort Express as a way of hiding his machine behind a NAT router when he's in a hotel or traveling. And Steve asked if there is Windows software to configure the AirPort Express. The answer is yes. Apple has created an AirPort configuration utility for Windows. It works just fine, in my experience. I've used it when I didn't have my MacBook Pro handy to fix someone's Time Machine base station. Apple in recent years has done a pretty good job creating Windows software and making their OS play nice. That's great news. Didn't know that.

Steve: Nor did I, so I wanted to make sure our listeners knew because the question did come up when we were talking about it before. It's like, well, would this work in a Windows-only environment. And Paul says, yes, absolutely.

Leo: Finally, our last question, from Amir Katz in Kfar Saba, Israel. He wonders about TPM, the Trusted Platform Module. Following up on your answer from Episode 200, how to activate TPM in the BIOS of your system, I wonder whether there are any downsides to doing that. I do recall your explanation how the DEP - Data Execution Prevention - was introduced in the x86 family, and it used to cause a lot of applications to crash as they were trying to execute code on the stack or heap. And DEP, of course, prevents that. So I wonder if TPM might have similar side effects, breaking some applications? If there are no issues, why don't BIOS vendors or the PC vendors turn it on by default? Also, apart from full-disk encryption, what other benefits do you get by enabling it? Thanks, Amir.

Steve: That's a great question. The answer is, do you remember how much trouble Intel got in...

Leo: Yes, yes. I know where you're going.

Steve: Uh-huh.

Leo: They announced processor IDs with the Pentium.

Steve: Yes, yes. The idea that every single chip would have a unique serial number. And it didn't take the privacy people half a millisecond to say, whoa, wait a minute. What? And in fact I remember going into BIOSes that had it turned on by default and turning it off. It's like, thank you, but I would prefer not to have, for whatever reason, it just seems like a bad idea for my Pentium to have a unique serial number which is available to the outside. And so the good news was, you were able to disable that feature, and now it's gone completely because it was just a bad idea.

Well, the Trusted Platform Module is very much the same. There's definitely a privacy-related concern because it has the ability, I mean, part of its function is to be an anchor for identity of the machine. I would argue that its benefits outweigh the liabilities. But the motherboard BIOS manufacturers weren't going to have this happen to them again. So they said, okay, just so that no one accuses us of having a serializing crypto unique machine of any kind on every single motherboard, we're going to just have it off by default. If the user needs those services, they can turn it on.

And to answer Amir's second question, apart from full-disk encryption what other benefits do you get, well, it is the identity enclosure for your system. There's actually a lot more that the TPM is doing than just providing sort of a vault for security information. During the boot process it's actually looking at the image of the code running during booting and performing hashes of it to make sure that step-by-step during the boot process nothing bad has happened. There's no malicious code that has managed to stick itself into the boot process. So it's very useful.

The idea is that, if you start from a known secure condition, that is, with the machine off, it's hard to get much more secure than that, and if you validate every stage of the booting process using secure validation technology, then once the OS gets booted, it can be sure that nothing malicious has happened before it got in control. And the Trusted Platform Module makes that, I mean, that whole process possible. So it's very useful. And of course it's also useful for providing secure authentication. If you're going to use a fingerprint reader, like at boot time, then it's always tied to a TPM chip. So you enable that, then you enable your fingerprint reader, and it's very possible then to swipe your fingerprint and to have that verified by the BIOS and have that provide a password to the hard drive that unlocks the physical hard drive that makes it possible to boot. And absent both the fingerprint swipe and the TPM, nothing can access the data on your hard drive.

So there's a lot of good stuff there. Not only hard drive encryption, but also authentication. And as we learned in Windows 7, they will be - and I'm really excited about this in Windows 7 - they will be making a public API that would, for example, allow applications running on Windows 7 to say, hey, before we do this, please reauthenticate yourself. That is, for example, it's a feature that I will definitely take advantage of in CryptoLink where you could require that the server ask a client to reverify that they're really themselves, from a human factor standpoint, by wiping their finger across a fingerprint reader. Right now there's no way to do that because there's no common API for the fingerprint reader. But that's something that Microsoft is going to be providing in Windows 7, which will be very cool. And again, it's all based on the TPM.

Leo: Yeah, I mean, I think it's - I remember a lot of controversy when TPM was first proposed, and people like Cory Doctorow were freaking out because one of the possible uses would be to expire documents, that you could send a Word document that would be expired after a certain period of time, I mean, there were all sorts of issues.

Steve: Right, well, because it's nonspoofable. It is deliberately a nonspoofable technology. So, sure, somebody could abuse the power that it gives them. But, you know, it's like, well, okay. They could also, I mean, there's all other kinds of ways they could do that, too.

Leo: And as a result I think manufacturers opted to, well, let's just avoid the whole thing and default to off. That way we don't have to defend it. And if somebody needs it or wants it and in full knowledge of the risks and benefits turns it on, all the better.

Steve: Right.

Leo: Steve, once again, a wonderful time. You get your questions in to Steve by going to GRC.com, that's his website. And if you go to GRC.com/feedback, that feedback form is right there. You can ask questions there. You can also get the 16KB versions for those of you who are bandwidth-challenged. We've got show notes, a full transcription by Elaine, a lot of detail. And of course SpinRite's there, the world's best, finest, one and only hard drive maintenance and recovery utility and all those great free programs.

Steve: And we even have a valid security certificate, Leo.

Leo: And a shiny new security certificate. That's GRC, Gibson Research Corporation, GRC.com. And that tells you how long Steve's been doing this. He's got a three-letter domain name, which are as rare as hens' teeth. Steve, great to talk to you, as always. I am in China as this airs. And so, let's see, I think our next live record date will be...

Steve: Two weeks from now.

Leo: Two weeks from now. It'll be...

Steve: We're going to do the mega security update podcast. It would normally be a Q&A. But we will have been not recording live at that point for three weeks.

Leo: There'll be lots to talk about.

Steve: Lord knows what will have happened. I hope the Internet is still here, computers are still booting. And then we'll let our listeners know what has happened in the three weeks that you've been floating around Asia.

Leo: You bet. You can watch us do this live Wednesdays, again, starting up again July 22nd, Wednesdays at, let's see, we start at 2:00 p.m. Eastern - that's 11:00 a.m. Pacific or 1800 UTC - at live.twit.tv. Or just listen to the podcast when it's available every Thursday on iTunes and the Zune Marketplace and everywhere.

Steve: Every Thursday like clockwork.

Leo: Like clockwork, thanks to Tony and now Erik, who's also working on the show. Thank you, Steve. We'll see you next time on Security Now!.

Steve: Thanks, Leo.