Listener Feedback #88

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-240.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-240-lq.mp3

Leo Laporte: This is Security Now! with Steve Gibson, Episode 240 for March 18, 2010: Your questions, Steve's answers #88.

It's time for Security Now!, the show that covers all your security needs online, your privacy, your protection. And the man who does it all for us, the great Steve Gibson, of GRC.com, the guy who created - actually discovered spyware, and created the name spyware, and wrote the first antispyware tool, and is the author of SpinRite and many other great utilities. Hey, Steve. Welcome.

Steve Gibson: Hey, Leo. Great to be with you again. Episode 240.

Leo: Yow.

Steve: Two four zero.

Leo: Now, we're back on the even shows, the mod 2 shows are Q&A. So we've got a Q&A today.

Steve: Yeah. And I've got to say once again, I mean, I know this is redundant. But when I read through our listeners' feedback, it's just so fun. I mean, we've got people who are paying attention and engaged and involved and - anyway, I love it, so.

Leo: Yeah. Well, you know, this is, as many have said before, a master class in, not just security, but lately in computer science. And so if you listen to this show you pay - you have to pay attention.

Steve: Somebody did write that I read, I don't know if it's one of our questions today, but I read his email this morning saying that he loved - he was loving the current computer series, "How Computers Work." And he says, "That's what makes this podcast different from any others."

Leo: Yes.

Steve: And he referred back to the "How Cryptography Works," the series we did that really explained that in detail carefully so that everyone could get it. And it's neat that it's not something you can find anywhere else.

Leo: Yeah, I really like how you've - you've now incorporated security news, so that you do hear what's going on in security. We're going to get that in just a little bit. But it's education as well as the news. And that's a perfect combination, if you ask me. So what is the latest security news?

Steve: Well, we don't have a super heavy week, which...

Leo: That's nice.

Steve: ...is good news. We do have, naturally, it's sort of the usual suspect. There's a new zero-day vulnerability in Internet Explorer. A problem was found in something called the iepeers.dll which permits unfortunately remote code execution. It's a DLL which is used to support print and web folders. And the exploit would be that the bad guys, as we've heard so often before, send out a URL in email or through whatever avenue, one way or another getting you to go to - getting your browser, IE, to display a page that was specifically designed to take advantage of this problem. The bad news is that it is zero-day, so the first that Microsoft learned of it was where they found out that this was being exploited in the wild, targeted attacks, being used for targeted attacks.

Leo: Now, let me ask you about zero-day. Because I always thought zero-day meant within 24 hours after Microsoft posting a patch. But it could be the other way around where we discovered it because somebody was using it, as in this case.

Steve: Right. The formal definition is that before it was known, it was in use. So the bad guys found something that was exploitable and were exploiting it. And it was through that that it came to the attention of the vendor. So it's like the reverse direction. So it's not just reverse engineering a patch which allows them to figure out what was wrong and then create malware for it, even though it's technically been patched, but then they're taking advantage of the patch delay that people to varying degrees have. Here it's actually in use when it first comes to the attention of the vendor. So that's sort of the

formal definition of a zero-day exploit.

Microsoft is not happy. They are now testing a patch. Okay, so not everything is affected. Windows 2000, XP, Server 2003, Vista, and Server 2008 are; but Windows 7 is not. So if you have moved to Windows 7, and you're still using IE, you're okay. Oddly, IE5 and 8 are not vulnerable, but IE v6 and 7 are. So...

Leo: How weird.

Steve: So they introduced the problem in IE6. And the changes made to IE8 keep it from being a problem. So, although no fix is available, I mean, I would say, if anyone who is listening is still using IE when it's so clear that Firefox is a safer, albeit not perfectly safe, but seems to be having less of these kinds of problems than IE does, certainly here's another reason to move to Firefox. Microsoft's workaround in the interim is you can disable scripting. So that's one solution.

They do have one of their OneClick Fix it solutions because there's some complicated registry edits that it's possible to accomplish which shuts down IE's ability to invoke this, although at the cost of some lack of functionality which you arguably might need. But if our listeners go to support.microsoft.com/kb/981374 - so again that's support.microsoft.com/kb/981374 - that takes you to Microsoft's quickie fix page, where there's just a button you can click. And it will use scripting in order to run their little - their what they call their Fix it solution, which will perform these edits for you. After this is patched, you'll need to go back and turn this back on if you do want this functionality. So this is our typical problem came out, Microsoft hasn't had a chance to fix it yet. It is being used in targeted attacks. So these are the kinds of attacks we're seeing more often now than we used to.

Leo: Spear phishing.

Steve: So, yes. And what was our term from last week? Weaponized email.

Leo: Oh, what a world.

Steve: And we have another term we'll be introducing here in a minute. Over on the Apple side, Safari has upgraded, or updated. Everybody should now be at 4.0.5.

Leo: Yeah, I got that yesterday. I noticed that.

Steve: And I got it just when I turned my Mac on in order to fire things up for the podcast. This fixes 16 security problems, not all security flaws. For example, one was a bug that allowed cookies to be set even when they were supposed to be blocked. That's something I've actually known about for a couple years due to my cookie research. And I had it listed on one of my web pages as here's a problem with Safari. I don't expect that I'm responsible for having pushed them to fix it. I think they finally found out about it, someone brought it to their attention or whatever.

Anyway, six of the 16 flaws only affect the Windows version of Safari. The other 10 of the 16 affect both Windows and Mac. So if you are a Safari user, you'll want to get that updated since they've got a bunch of things patched. And I'm not going to go through in detail what they are. But they're remote code execution or application termination - which is a nice way of saying "crash" - bugs which Apple has now fixed.

And I did want to note that, just in passing, the news this week that Twitter has introduced link filtering, which they're very excited about. In one of their blog postings they talked about trust and safety. This has been a problem for Twitter because it's been possible for the bad guys to propagate malicious links using Twitter's broadcasting technology and catch a lot of people. So now what'll happen is, when links are pushed, they're using a redirection approach, sort of like we've seen with the standard link shortening, link redirectors like SnipURL and TinyURL. And in fact they are using, I think it's twi.tl is the domain that their little link redirector uses. And they point out that this allows them to fix things even retroactively. That is, if links go out which are later found to be malicious, since essentially all the links that they're sending route through their server before they redirect the person who's clicking them to the destination, they're able to, after the fact, block that and bring up a warning page telling people that we believe this is going to take them somewhere bad, are they sure they want to go further. So that's good.

In terms of a new acronym, I'm seeing - and this really was popularized at this year's RSA conference, the big annual security conference a couple weeks ago. The new acronym is APT, which stands for Advanced Persistent Threats. That's essentially what these attacks on Google and Adobe and the handful of other companies which now was believed to be launched from somewhere originally, to originate from somewhere in China. I would call them "long-term targeted stealth infiltration." That's essentially what this means. We're calling them Advanced Persistent Threats, meaning that someone somewhere, bad guys, were deliberately focused on penetrating a given single or multiple institutions, that they found some way in.

And remember, this is the big problem with security is it has to be perfect. Meaning it only takes one mistake somewhere, one thing missed, one vulnerability not patched, one port left open, one unsafe application running. I mean, literally, the barrier is so high to be absolutely secure because it just takes one hole for some guy to get in. And so if there's tremendous pressure against the security perimeter, any leak will allow someone in.

So this notion of an Advanced Persistent Threat is that some way in is found, and then the bad guys set up a persistent presence inside the network and attempt to stay undetected and connected in the network, present essentially, for as long as possible, for doing whatever they're doing - surveillance, collecting files, sending them offsite, out of that local country zone, wherever.

One of the security testing services, NSS Labs, created a slight variant of the original Aurora exploit which was used to get into Google and Adobe and 20-some-odd other organizations, and tested six major AV products against their slightly altered exploit. All of these products - Eset, Kaspersky, Symantec, Sophos, AVG, Trend Micro, and McAfee - all of them had been updated to catch the original Aurora threat. But obviously they didn't catch it before it was known, which of course is the problem with signature-based AV filtering. But when NSS Labs made a slight variation to it, of those six AV products, only McAfee happened to also pick up the variation. The other ones said, well, that's different than Aurora, so we're going to let it through. It caused some concern because these other guys said, well, this is not the same.

And the problem is that - and this came up at the RSA Conference, and I thought this was really interesting. The point was made that the bad guys who are creating this malware have the same access to the antimalware products as all the people using the antimalware products. So the bad guys know when they release some malicious agent into the wild that it's not going to be seen because they're able to, and they do, test it against all of the existing antimalware programs.

Leo: I never really thought about that. But of course they can do that.

Steve: Yes. We've never talked about this. I thought that was a really, really salient point because - so they release it. They know it's not going to be seen. And so this created some discussion about, well, how do we, like, arrange to have our antimalware products be smarter about trying to target these things? And the problem is that then you start heading toward what we have talked about in the past, and that is sort of more behavior-based screening where you look a little bit, not just strictly signature based, but you look more at, like, okay, here's a vulnerability. So what would software have to do in order to exploit this known vulnerability? The idea being you try to design something which, without seeing a sample of what's bad, you design something that ought to catch anything that tries to take advantage of that particular vulnerability. And what immediately happens is your false-positive rate jumps.

Leo: Right.

Steve: You start seeing things that are benign which you cannot differentiate from things that don't yet exist, but might be bad.

Leo: And as somebody who has been bit by false-positives, Steve, you know that that's not a good thing.

Steve: Well, it just doesn't fly. In a corporate environment where these things are deployed, suddenly you've got your antispyware technology telling you that known good things are a problem. And, I mean, nothing will upset users and force IT to back down faster than blocking good stuff, erroneously blocking good stuff. So it just, I mean, it really is a problem. It's - and I don't see any kind of an effective workaround solution for that. I mean, the notion that the bad guys have access to the same antimalware technology as the people trying to protect themselves is very powerful for knowing that these things will initially penetrate existing defenses.

And one of the strong points made at RSA was that we need something better. We need better defenses than we have because right now bad things are getting through. Now, it is the case that all of these exploits are exploiting mistakes, that is, we're past the point where people are doing really dumb things about security where, like, just leaving FTP servers with default passwords open. I mean, those days are gone. We're spending millions, tens of millions of dollars on security, yet major corporations are still getting infected. There was one botnet that infected 50 of the top 100. 50 of the Fortune 100 companies were infected with a single family of bots that were being remotely controlled because none of the antivirus, antimalware software they were using was effective against these things.

Leo: Amazing.

Steve: So it's a problem. I got a really interesting sort of fun email that I discovered just this morning when I was going through the mailbag, that I wanted to share. Sort of it ends up being about SpinRite, but it's sort of an interesting profile of one of our listeners, a doctor named Kent Bullis. He said, "Dear Steve and Leo, I'm a 51-year-old M.D. I started tearing stuff apart around age three. At five I told my mom I didn't have time to go to kindergarten. I was too busy. By 10 or so I started to fix small appliances around the house, so Dad wasn't quite as aggravated with me when he saw me with a screwdriver. I spent the summer of my freshman year in college building an A/D interface from the street components and 7,400 family TTL ICs, and programming a PDP-8, of all things, to collect, store, and manipulate solar insulation data from instruments on the roof, while the "lucky" college senior I shared the lab with got to play with a brand new 8080 chip.

"Fast-forward to medical school. Not smart enough or energetic enough to maintain my interest in computers at the same time, I was going through med school and residency and started my own practice from scratch. A few more years into being in charge of implementing an electronic medical records system for our university student health center, as well as participating on the implementation team of a large hospital system, all the while concerned about the security of our wireless network in the midst of 18,000 bright and energetic college students.

"I discovered Security Now! about two years ago, listen faithfully, love the show, and am constantly bugging our IT guys and double-checking what they're doing. They take it pretty well and admit I have raised a few good points. Listening to Steve talk about the PDP-8 in assembly language has taken me back and has been great. I bought a copy of SpinRite in '06, after I saw the segment Leo did with Steve about it. I just thought it seemed like a good thing to do. And I run it regularly on my own drives.

"So that brings me to my SpinRite story. This past Christmas I finally convinced my mother and sister, who live together, to switch from dialup to Comcast Digital. In the process of setting them up, I spent a lot of time trying to implement some improved security practices for them which I'd heard about on Security Now!. But when I got done, the machine wouldn't boot. After the password screen it went to an error message that a system file of some kind or another could not be found. My first thought was that someone had changed the password, but everyone denied doing any such thing. And then I realized the error message was not that of an incorrect password. So I tried to do a restore, but no luck. I got to thinking about all the times Leo has suggested that a system failure might be due to a corrupted hard drive."

Leo: I learned that from you.

Steve: "And I returned the next day with my copy of SpinRite. A few hours later, SpinRite was done, and the system responded to the old password and booted normally. Over the next few days I made a few visits to make sure my sister had followed my advice regarding creating a backup, and that the backup completed normally. I find common ground with you on several levels and enjoy the show immensely. I hope you have the energy to continue it for a long time and look forward to your finishing your secure wireless product. Sincerely, Kent Bullis, M.D., Medical Director, Ball State University Student Health Center, Muncie, Indiana."

Leo: Great story.

Steve: So neat note from a listener, yeah.

Leo: As always. We get - it really is nice, isn't it? It just feels good.

Steve: We've, yeah, got the greatest listeners.

Leo: Great to have the fans. So we have actually a truncated number of questions for you because I know that some of them are going to be in-depth.

Steve: Well, we have eight. And as I was thinking of the things I wanted to talk about, stimulated by the questions, I thought, okay, this is going to take a while. So, yeah, I thought we'll be a little deeper, we'll go deep today rather than have quite as much breadth as usual.

Leo: Yeah, I like that. I think that's great. That's one of the things that makes this show so good. Well, let's start with Andy Hamilton in Bristol, England and many other listeners who wrote the same thing. Apparently locking down Internet Explorer can break Firefox. Duh, what?

Steve: Yup.

Leo: Steve, hey from the U.K. I'm a SpinRite customer, and I follow you weekly on the Security Now! podcast. A few episodes ago you described locking down Internet Explorer, as it seems to have a positive influence even during non-related activities, you know, things like using Outlook email. Well, I turned up all zones to highest security, as you recommended. Shortly after, I noticed Firefox downloads of EXE files (from Dell and VMware) returned a zero-byte download, but with no errors or warnings. It turns out Internet Explorer is to blame. In Internet Explorer you have to go to Internet Options -> Advanced -> Reset. This returns the security levels for all zones to their default levels, compared to what I had them at, and then downloading in Firefox works. I can presumably turn them up slightly with some careful testing. What's going on there? That doesn't seem right.

Steve: I know. I have not tracked down the interaction, but the same thing happened to me and to a disturbing number of listeners. I would say "disturbing" except that I'm pleased everyone was interested enough to give this a try.

Leo: Does it do the same thing to Chrome?

Steve: I noticed it with Firefox. And in the little bit of exploring I've done, I think there may be a workaround. That is, there may be a way to keep the Internet zone security

turned up and still get Firefox to work. Let me back off a little bit and just remind our listeners that the idea was that Internet Explorer, which as we're pretty much constantly saying is substantially, unfortunately, less secure than alternative browsers like Firefox - which I'm now using and you're using and we recommend - and that it's not enough simply not to use Internet Explorer for surfing the web because, because of the integrated nature of Internet Explorer in Windows, anytime Windows is using a browser window in an application like, for example, in Outlook, which you may be using for email, that's the IE browser essentially being instanced into that window.

So because we're seeing such a problem now with, as we're now calling it in the industry, "weaponized email," where things are sent that look very reasonable, in some cases emails being crafted specifically for the target recipient so that they will - the facts will look right, it'll look legitimate, doing anything they can to induce them to click on that link. So the idea was to, even if you're not using Internet Explorer full-time, to lock it down so that incidental and even accidental use of Internet Explorer will be safer. So what that means is that Internet Explorer, which has this notion of zones, if you crank the Internet zone security to high, it really does prevent virtually all of these exploitations.

There's a side effect, though, which this listener and many of our listeners discovered, meaning that a lot of people tried this, and that is that for some strange reason you get zero-byte-length EXEs when you download them. And so it was shortly after I followed my own advice that I was trying to download some things, and they were coming in as zero-byte-length EXEs. And I figured, after some trial and error, okay, it must be the changes to IE. It's strange that changing IE's security settings affects Firefox's ability to download things. And I think there are some things that can be done over on the Firefox side. Anyway, I will have an updated solution, or none. But I'm going to pursue this, and I'll get an answer for our listeners, hopefully by next week.

Leo: It's really an odd kind of a thing to have happen.

Steve: And I did want to acknowledge everybody who wrote because a lot of people discovered this. And so, and if you've been having this problem and haven't figured out why, now you know why. You can go to put IE back where it was before with Options -> Advanced -> Reset, which will put all of your zones back.

Leo: But of course I don't want to do that.

Steve: No. I mean, again, we're dropping our drawers, unfortunately, when we do that. So not safe. I'm going to see - I looked at some blog postings, and there were some people reporting other ways to fix this problem. Now, I don't know if that's due to something else they had done in Firefox rather than changing IE. So I need to track down exactly what's going on, and we'll see if we can come up with a workaround. Because it would be nice to be able to lock down IE, yet still be able to use Firefox.

Leo: Might be a reason to use Chrome.

Steve: It took a lot to get me to go from IE to Firefox. So I'm hoping to stay on Firefox.

Leo: Yeah. You're a little behind. But actually I'm starting to prefer Chrome because it's faster than Firefox. Firefox has started to feel like IE. It's starting to be bigger and heavier.

Steve: I will say that, yes, that when you first launch Firefox, it takes a while to get it going. I mean, I don't shut it down and start it up during the day very much. I just sort of have mine - Firefox is running as an appliance sort of continuously on one screen. It's just my web browser screen. And then it's manageable.

Leo: I'm loving Chrome.

Steve: Really.

Leo: Yeah. But, see, this will be an - I'll have to do this experiment because, if it is - in fact, I'll do it, I'll run out, I have a laptop in my car, and I'll run out during the next question and get it. If it is in fact a global Windows protection, then it should affect every browser. Or if it's a weird interaction between Firefox, then Chrome would be immune.

Steve: Why has Apple got Safari on Windows?

Leo: God only knows. It makes no sense. There's no need for it. It's another browser. But you know there's a dozen browsers out there that we've never heard of, like Sleipnir and all sorts of weird browsers out there.

Steve: Well, but a lot of them are just rewrapping...

Leo: Right.

Steve: ...existing browser technology, like they're taking the IE control or the Firefox guts and putting different windows...

Leo: Well, in a way that's what Safari is because it's using WebKit, as is Chrome. So there's a number of WebKit-based browsers. And WebKit's really good. WebKit is an open source project that was based on originally the K-Browser on Linux.

Steve: Well, and I can see Google's impetus because they want to - Google is browser-based technology. So they want to establish their client on different platforms. And they've really gone out of their way to create a worthwhile and security-extended solution. But Safari on Windows is just, I mean, I'm...

Leo: It's superfluous.

Steve: Yeah, exactly. I mean, I'm glad for it only inasmuch as I'd like Apple to really have a really good browser because I'll be using it on the iPad here on Saturday, April 3rd, when mine arrives. You saw my note that I ordered it about two minutes after it went live.

Leo: As did I. In fact, we ordered - you know you can only order two. We ordered four because I have to give some away. So I'm giving two away. I wanted one for myself, and we're going to have one to have around the office for people to try. But did you order - now, let me ask, did you order the basic 16GB version? Or did you order the 64GB?

Steve: Cheapest one they had.

Leo: Me, too. I feel like 16GB is plenty.

Steve: It's plenty. It's the best value. I mean, in terms - this is not the one Apple wants you to order because...

Leo: Probably not, yeah.

Steve: ...they're making the least money on it. My theory is, I'm not sure I'm going to use it, so I didn't want to throw money away. 16GB, I mean, we're so spoiled now.

Leo: I know. Oh, I need 64. No. And if you're not going to put music and movies on it, you don't need it. I mean, if you start loading it up with movies - but even then, 16GB is like 16 movies. That's a lot of movies.

Steve: It's really - yes, exactly. Exactly. And my other thought is that a year from now - we already know Apple's track record.

Leo: Right.

Steve: They're going to add the camera, they're going to add the GPS, they're going to add all the things that it didn't have initially, and they're going to lower the price. So it's like, okay, that's the one I want, the one that they haven't announced yet and won't be available for a year. And so if I end up really using it - and I really think I do. I think it'll be my PDF reader because I don't have a good, a real portable PDF reader, and a good connection to the Internet, a good web browser.

Leo: Right. And for that 16 is plenty.

Steve: And I'll use my little MiFi gizmo in order to link me. So I did not get the...

Leo: Same thing.

Steve: Besides, I'm not using AT&T for anything, so...

Leo: Same thing. And we'll use Verizon's MiFi.

Steve: Yup.

Leo: Have Verizon surfing speed. Now, there's some question about whether, you know, they have this $30 unlimited AT&T. And there's some question. And nobody is saying that that's 5GB unlimited. It may in fact be truly unlimited, which would be interesting.

Steve: Well, unless you're in New York or San Francisco.

Leo: And then you can't use it at all.

Steve: In which case it doesn't work at all. Because, I mean, because AT&T's network has just collapsed under the iPhone load.

Leo: It was interesting, you know, that happened at South by Southwest last year. So AT&T brought in trucks, I saw one, with towers. They brought in additional cell towers. And it worked flawlessly, even with - there were 40 percent more people at South by Southwest this year than last year. There were 17,000 geeks. And all of them had, well, I have to say, last year all of them had iPhones. This year there were a lot of Motorola Droids out there, quite a few Nexus Ones. So maybe that's another reason why AT&T did better. Next question.

[Talking simultaneously]

Leo: Go ahead.

Steve: No.

Leo: Mark in Melbourne, Australia reports a problem with lockdown of IE when in

Sandboxie: Hi, Steve. Long-time listener of Security Now!, sometime emailer. I'm very appreciative of all you and Leo do. Keep up the great work. Thank you. I'm a couple of podcasts behind, so you may have already covered this. But just in case, on a recent episode of Security Now! you gave instructions on how to lock down IE, even for those using other browsers, by locking down the IE trusted zones and only allowing, for example, Microsoft updates. This is what we were talking about.

Since it seemed like great advice, I acted quickly on it. However, I found another small problem. My file downloads in Firefox stopped working kind of. As I mentioned in the subject line, I run my Firefox browser in Sandboxie. And I run NoScript, but don't call me paranoid. No, you're just Steve. And I found that each time I clicked any download link it would fire up the Firefox Download Manager, as usual, but then the download immediately got canceled. I then tried to click the download manager to refresh/retry, and it seems to work, and Sandboxie throws up the usual "Do you want to recover this file" dialogue. Then I click yes, and the resulting recovered file would be zero bytes, as we were talking about.

After poking around for a while to try to figure out what was going wrong, I finally remembered that I had recently locked down IE as per your instructions. I reversed the lockdown. Lo and behold, my Firefox download/Sandboxie recover started working again. Go figure.

Just as an aside, I thought it'd be good to get another plug-in for Sandboxie. With all the recent talk in Security Now! lately about Flash and browser vulnerabilities and talk about doing banking on a different machine, et cetera, might I just say that, when I set up my most recent machine, I built a fresh install of Firefox, and then only ever run it in one of two sandboxes: one, for my banking, with clear on exit; or, two, for ordinary browsing, which I periodically clear but retain history and bookmarks. When I see an update is available I exit, clear my sandboxes, and run Firefox outside the sandbox just long enough to update it. Seems like a good security setup. What do you think?

Steve: Well, I like this because of course he talks about what we were just talking about, about the problem with locking down IE and its interaction with Firefox. And but I did agree with him that reminding people about Sandboxie was a good thing. When he talks about doing this file recovery, what he refers to is that's the way in Sandboxie of manually and deliberately pulling a file out of the sandbox into your main system. So he was saying that similarly, when IE was locked down, it was affecting Firefox even inside the sandbox. So his approach, which normally worked, of extracting a file from the sandbox when he wanted to keep it permanent on his system, was failing in the same way that it was for people not running Firefox inside the sandbox.

But he asks about the use of the sandbox from a security standpoint for protecting his banking and for ordinary browsing. And he's doing something good. I like this idea that he uses one sandbox for banking, which he clears on exit, so that prevents that instance of the sandbox from accumulating anything that's bad. The problem here is again, looking as we always must at the worst-case attack from a security standpoint, the sandboxed data is still moving through his machine, that is, his Windows machine on its way to his bank. So a problem on the outside of his Windows machine could affect the security of what he's doing inside the sandbox. The flipside of that is that, if the sandbox works correctly, it really should, and it certainly has been designed to, prevent problems that our browser originated from ever escaping the sandbox.

So while it's worth bringing up the caveat that, if his external Windows machine had become infected, that could compromise the security of what he's doing in the sandbox, that said, the use of Sandboxie should and probably would contain any problems that occurred. So I do think it's much better and much tighter security than not using the sandbox. And of course having NoScript on top of that is belt and suspenders. And I don't think I would call him paranoid. I think I'd call him a good Security Now! listener.

Leo: A wise man.

Steve: Yeah.

Leo: Doing what Steve does, probably.

Steve: Being very safe, yes.

Leo: Question 3, Barry Ardolf in Minneapolis, Minnesota wonders how to block 84.124.5.162. Hello, guys. I've been a victim of malicious code many times, and I trace it to sites with no DNS translators. How do I block straight sites which do not go to a DNS? Here are the examples of what I need help blocking. Do not go to these sites as you will get infected, folks: 84.124.7 - I'm not going to give you the whole address.

Steve: Actually I changed the IPs.

Leo: It looks like you did.

Steve: He gave them to me literally, and I said, oh, no.

Leo: Looks like you did.

Steve: Leo might read these, so we're not going to, I mean, who cares what they are?

Leo: Right. Also - so first of all, he wants to know how to block by number, not name. Also, how does one complain to authorities about these sites so they get taken down? The account complaint address, as an example, abuse@yahoo.com, comes back as rejected, as in we all know the site lies about itself when it registers. So please educate the world on how to block sites like this. Routers and host files cannot block straight numbers. I like your podcasts. Please talk more about Windows 7. I have not been required to use SpinRite, but I know about it.

Steve: Okay. So a couple things here. It is the case that the hosts file cannot be used to block direct IP-addressed access. Remember that the hosts file is sort of a - it's sort of a proxy for DNS, so that if something is listed in the system's hosts file, your system will

go check that before it goes to whatever DNS servers are registered for the system. So that's an effective way, for example, if you put DoubleClick.net in your hosts file, then your system will simply not get to DoubleClick.net, period, because it will always look in the hosts file first. But it does not work for IP addresses. The system sees it's an IP address, doesn't do a DNS lookup on the IP because that's, after all, what DNS provides.

So the question is, how do you block simple IP addresses? Well, you'd have to check the specifics of whatever personal firewall you were using. But many personal firewalls do allow you to put individual IP addresses in, in some cases even with wildcards, in order to block a single IP address or a range of IP addresses. I know that the older time firewalls, back in the early days of ZoneAlarm and many of the firewalls as they were originally being created, did allow you to manually enter IP addresses. So if you wanted to have that kind of functionality, that's the best way to do it.

There is some technology in Windows which could technically use a very lightweight client in order to perform this kind of blocking. But I'm not aware of anything that's as compatible as just choosing a personal firewall and using it for outbound control, which is to this day still a feature that is available from any of the personal firewalls, but still not from Windows.

Leo: Windows Firewall won't let you do that.

Steve: No.

Leo: Weird.

Steve: Now, I've heard, I have not looked at it yet because I'm not yet a full-time Windows 7 user, but I did hear that Windows 7 had begun offering outbound blocking technology, though I've not looked at it closely. Now, as to how you complain to authorities, here's the problem. You have an IP address, and it lacks any DNS information. If it's a malicious IP, it's very likely not registered either with forward DNS that would cause it to get looked up or even reverse DNS that would allow you to turn the IP address back into a domain. And even if it were, it might be bogus information.

Mark Thompson, a good friend of mine, founder and creator/proprietor of AnalogX.com, created a really interesting site that I've referred to a couple times over the years called FixedOrbit. I have no idea why he called it that. It's FixedOrbit.com. That's his site. What FixedOrbit does is extremely cool, and as far as I know it's unique. It goes out onto the 'Net and pulls the routing tables from a cross-section of main major routers on the Internet and pulls them into a database, processes them, and ends up producing essentially an analysis of who owns what IPs.

So there's a Tools tab that you can choose on the main FixedOrbit page which allows you to put in any IP address, and it will map that back to the hierarchy of networks that contain and own that IP address. That allows you to just give it an IP that you know nothing else about. It allows you to figure out who controls that. And from that information you can probably track them down and find an administrator at some level to report the problem to.

So it's actually the best tool I know of for, when there's no other association to just a random-looking IP, of figuring out where it is geographically, what hierarchy of networks

contain it. When I talk about a hierarchy, I mean that many times, like a Tier 1 ISP will own huge, huge blocks of IP addresses. But then they resell subsets of their large IP network to ISPs, who may then in turn sell subsets of their network to large customers and so on. So for a given IP address you actually have sort of a hierarchy of ownership.

And you can do, I mean, if you wanted to you could complain to the Tier 1 provider who's at the highest level. They're probably less concerned about individual problems, but they would have some motivation to deal with something that was malicious. So you probably want to find someone as close to the IP in terms of the hierarchy as possible because they would have control over that particular IP address and say, look, here's a problem. If you don't get any response from them, you can sort of move back up the hierarchy and say, hey, here's a problem with one of your customers who's hosting an IP address that's malicious and so forth. Or you can just block it and not worry about it. I'm not sure what sort of traction you get these days complaining about this kind of problem because it's just so prevalent.

Leo: Well, now I've got some crazy results for you. First of all, the Windows 7 firewall does in fact include both inbound and outbound. In fact, you will be, I think, impressed.

Steve: Yay.

Leo: It is, wow. It, like, is a full-featured firewall. So I was stunned. I actually hadn't looked at it yet.

Steve: Very cool.

Leo: And I turned up my security, and this is on a Windows 7 machine, all the way up to high on Internet, all the way up to high on local Intranet. I added only two trusted sites. Those were Microsoft.com and WindowsUpdate.com. And I was able to download with no problem Wizmo.exe in Firefox. And there you see it running. So I don't actually...

Steve: Interesting. I'm on XP.

Leo: Yeah, I wonder if it's an earlier version of Windows issue because there's the download. And here it is...

Steve: And it works.

Leo: Yeah.

Steve: And it's not zero bytes.

Leo: No, it's running. I don't think it would run if it were zero bytes.

Steve: My code is not quite that lean, Leo.

Leo: It's lean. Well, it's only 38K, so it's a good test for this.

Steve: Yeah.

Leo: Because it's so small. But, yeah, it ran fine. So I was prepared to go to Chrome to see if it worked, but I didn't have to because it worked fine in Firefox. That's the latest version of Firefox and the latest version of Windows 7 with all updates. So I'm not sure which machines are having that problem.

Steve: I'll figure it out. I'll figure out what's going on.

Leo: Sounds to me like maybe it's an older version of Windows problem. And Windows, by the way, Windows 7 Firewall, I think you should take a look at it. All right, moving on to Question 4. G. Wade Johnson in Houston, Texas comments about write-only programming languages: Steve, I've been enjoying Security Now! for a while. Particularly I like the way you carefully explain concepts that may be new to your listeners. I've been a professional programmer for a number of years in several languages and find that many of your programming comments really match my experience.

However, I recently listened to Episode 236, where you made a comment that I had to disagree with. After talking about programming in assembler, C, and Perl, you stated that Forth really was a write-only language, meaning that it was easy to write, but difficult or impossible later to read. Many moons ago I was actually a professional Forth programmer. For eight years I helped develop and maintain a very large codebase in Forth. It was no more unreadable than code I've maintained in C, C++, Java, or Perl, before or since that time. While it is possible to write unreadable code in Forth, that's true about any programming language. Given some good design skills, you can write truly elegant code in Forth - that's kind of been my experience, too - much like it's possible to write elegant assembler, I would bet.

Throughout my career I have regularly heard people claim languages that I've used were write-only - Perl, C, C++, Lisp, et cetera. In every case I've also seen really clean and readable code in each of those languages. I suspect most of the problem has to do with the commenter not being familiar with the idioms of the language, rather than a failing of the language itself. Sorry for the rant. Keep up the good work. I really am enjoying the "computer from the ground-up" series. It takes me back.

Now, Forth, the problem with Forth is it doesn't have much of an idiom. The idiom is created as you create the dictionary. But if you do it sensibly, it looks like English.

Steve: I guess I'll temper my comment. And I thought a lot about why I'm feeling that

where he's not. And I guess so what I'll say is that I do think it's fair to say that different languages encourage and facilitate different degrees of, I'll call it transparency. For example, my experience with Pascal has been that I can come back long after I wrote some Pascal, and it just seems so clear to me what I did.

Leo: Well, yeah, but Niklaus Wirth designed that as a teaching language. So it was intended that way.

Steve: Which is my point.

Leo: Right.

Steve: Is that it succeeds in just - in having - in being transparent. And when I'm thinking about trying to understand some Forth code that someone else wrote, the problem for me is that there is, well, the problem is the elegance and the power of Forth is its use of the stack. So we ought to talk a little bit about Forth and the stack because that was the context in which I mentioned Forth last week also was when we were talking about the whole concept of a stack. In Forth, you don't have arithmetic expressions that are algebraic the way you often have in other languages, where you say A equals five plus four. Instead you push five on the stack, and then you push four on the stack, and then you do a plus sign, which adds the top two things on the stack, leaving the sum of them on the stack. And it's very powerful in the way that, like, the early HP calculators, which used RPN, Reverse Polish Notation, which were also stack-based calculators. They were very powerful.

But for me, you can't look at Forth code and see what's going on. You have to follow along with Forth code because there's this hidden state, that is, the current state, the current contents of Forth's execution stack completely affects the result of the verbs which you are using to apply against the stack. So, I mean, I respect a professional Forth programmer, and I will take our listener's word for the fact that, if you embed yourself in Forth long enough, you can read it. I find it difficult to read, but I have not spent any great length of time programming in Forth. I think it's interesting, and I learned it and used it for a while.

And as you say, Leo, the nature of, at a much higher level, viewing Forth at a much higher level, the way you create - you essentially sort of build your own little environment by creating a meta language with your own verbs out of the lower level intrinsic language and verbs in Forth. And it's really neat. I mean, it's an interesting development environment, unlike anything else. And by the way, if any listeners are curious, there's all kinds of free Forths around for all the different platforms. I mean, it's an interesting enough language that it's been implemented many times on all kinds of processors. So it's...

Leo: It's also very spare. It's small.

Steve: Yes. And in fact that's one of the reasons - in fact, is it Sun? Somebody uses Forth to boot their machines.

Leo: Oh, really.

Steve: Yeah. Forth is used as, like, the BIOS language.

Leo: Yeah. It's a great embedded language. You can have a Forth interpreter in a few K, maybe even less. And then of course everything that Forth does is in a dictionary. So it can be very, very small. It was written for telescopes. It was written to control telescopes by Charles Moore.

Steve: Yes.

Leo: I interviewed him when we were at Tech TV. And he was stunned that I even knew who he was or what Forth was. It was - poor guy, I mean, he really created an amazing thing, which is still I think used in robotics a little bit. I'm sure it must be. Wonderful language.

Steve: Yeah. It has not died. And as you say, if someone were to create a new chip, and with a random instruction set, and needed to quickly get something up and going, one of the quickest ways to bootstrap a system is to write a small Forth interpreter and then start writing Forth code. Especially if you've got a body of Forth code you want to immediately port over. You can get it up and running on an arbitrary architecture very quickly. But still, I would - my feeling is it's the fact that you have to follow along with the code to track in your mind what's currently on the stack, and that that's a completely opaque thing. You can't see that in the language. You have to execute the language in your mind in order to see what's on the stack, to know what's going on.

To me that's very different than looking at a language like Pascal or even like C, where there isn't anything hidden based on, I mean, I guess the contents of variables would be in a sense hidden. But to me you're seeing it - I guess it's just that I'm not used to Forth. But you see what I mean, that there is this state of the Forth machine which the language affects. And you have to know what that is in order to be able to read the code. So I think it's a little different.

Leo: Question 5. I'm sorry. Is it already Question 5?

Steve: Yeah.

Leo: Yeah, Question 5. Greg W. in Brisbane, Australia - another Australian, that's what confused me - plowed into the Blizzard Authenticator hack. He said: Steve and Leo, after listening to your feedback podcast where you discuss a question regarding the hacking of the Blizzard Authenticator, I figured I'd do a little more research. He went to NetworkUptime.com and read an article. We've posted that in the show notes. To get the gist of the protocol, and learning that the password is hashed with a salt and not sent in the clear, I fired up Wireshark, which is an Ethernet monitoring program, right, you monitor the traffic...

Steve: Exactly.

Leo: ...that's crossing the Ethernet - to see if the authenticator code was sent in the clear, as well. I discovered that by typing random numbers as well as correct numbers, the packets never contained the authenticator code in plain view. So far so good. This sums up what others have noted, that the man-in-the-middle malware is a DLL that sits on your PC, waiting for you to launch World of Warcraft.

Then it wakes up, keylogs your password and authenticator code, and does two things: First, it sends the valid data to the hacker, who only has a very small window of opportunity, obviously, to use that information. Then, secondly, it sends an incorrect authentication code along with the password into the WoW client to pass to the real Blizzard logon server, which then correctly refuses the logon. This little keylogging nightmare stays in the PC and is smart enough to prevent the user from accessing either the game itself or the Battle.Net Account Management pages. It probably sniffs those as well and uses the login to futz around with the victim's account details, although the CC number, the credit card number, or the user's full name and address can't be obtained, nor can the authenticator be removed without knowing its serial number.

Quite a piece of work for hacking into someone's game account rather than banks. Although hacking WoW accounts is a serious, illegitimate business, even as banks in the U.K. are discovering that stolen credit cards are being used more and more to create "legitimate" WoW accounts to spam the legitimate users in-game. And isn't that amazing? What is it about WoW that attracts these bad guys?

Steve: Well, I liked and I appreciated Greg's report because it gives us a snapshot into a perfect example of the reason I brought this up last time, and that is that even though we've got a one-time password technology, here's an example of that not being enough. That is, you know, this Blizzard Authenticator is the PayPal football. I mean, it's exactly the same thing. It's using the one-time password technology, which is time-based, to authenticate against Blizzard's authentication servers. So that's why he mentions that there's only a short time window in which this can be used, because he understands that the football, which is time based, and we've talked about that extensively, the same football that PayPal uses, is only valid for a short period of time. I think it changes every 30 seconds to a different code, although as we remember that first digit of the six digits does provide some synchronization information in order to make the system more tolerant. So even despite all this, if you've got something in your machine which is monitoring, this is a man-in-the-middle attack by a piece of malware which is catching the one-time password and forwarding it in real-time to somebody somewhere who then logs in as you, impersonating you, even though this is specifically what this technology was designed to prevent.

The only reason you would use the Blizzard Authenticator is that you were trying to protect yourself from keystroke logging. But that would be persistent keystroke logging. Here we've got on-the-fly keystroke logging, which is intercepting the valid data, replacing it with invalid data so that the person trying to log on isn't able to, which allows the bad guy the opportunity to do that in his place. So this is a perfect example, and I like the details, of how something that is specifically designed to prevent someone from being able to log in by impersonating you doesn't do the job because it isn't safe against man-in-the-middle attacks.

Leo: It's amazing. It's really quite a clever hack.

Steve: Yeah.

Leo: Question 6, as we move along through the list. Luke in Boston, Massachusetts asks: Why a language virtual machine? Steve, I've really enjoyed your current "how to build a computer" episodes. Your detailed discussion of how computers work in hardware has made me wonder what the story is with virtual machines, in particular the virtual machines that show up in various language implementations. I understand how a full system VM like VMware works, and what the value is. But I'm not sure what a language virtual machine, like a JVM, the Java Virtual Machine, or Google's V8, or Parrot, which is the new platform for Perl 3, is, or why - I'm sorry, Perl 6 - or why a language designer would want one.

It's all about cross-platform portability. Is it all about cross-platform portability, or are there other benefits? Are language VMs just software simulations of a particular existing chip, or at least simulations of a chip that could exist? This is - I love this question. Or if not, and the instruction set has surpassed what could be done with the transistor logic you described in Episode 233, why are they virtual machines instead of just programs that read other programs? Thanks for making such a great show. Luke. Well, you mentioned Pascal. Pascal was originally a P machine.

Steve: Yes, p-code was what the compiler produced. It is the case that a language virtual machine is different, as he says, like for example from a VMware system where we're virtualizing an operating system, so to sort of create a containment or a duplicate of the operating system. In a language virtual machine the virtual machine is designed for any of a number of different purposes. It is normally a sort of an idealized processor, that is, you would, if you were implementing a language like, well, we'll take the example of Pascal, but Java is a good one, too. You're implementing a language.

So as a compiler designer you can say, gee, wouldn't it be great if the processor that we were running on, we were able to just, like, design our own, that it had the following characteristics. And so the virtual machine is created to emulate that environment. I don't want to say a processor because it could sort of be more than a processor. So the virtual machine is, exactly as the name says, a virtual machine. It is an emulated pseudo computer which doesn't necessarily exist in hardware. But, for example, there actually was a Forth chip created which implemented the Forth language virtual machine in hardware. So it is possible, in fact I think there were some p-machines that were implemented in hardware, too, so...

Leo: Yeah, I think so, yeah.

Steve: Yeah. So you can, if you wanted to, devirtualize the virtual machine, make it a real machine, and you get all kinds of speed advantages if you were to do that.

Leo: This might go back to Donald Knuth, who wrote his classic books on programming in a pseudo-language called...

Steve: MIX.

Leo: ...MIX.

Steve: MIX was what Don created, yes.

Leo: Because he didn't want to make it machine dependent, I guess.

Steve: Well, he wanted his books to be able to survive through the years.

Leo: Smart man.

Steve: And he also wanted to illustrate his concepts in actual instructions, not just sort of an algebraic abstraction. So he had to create something that the students could read. In the inside front covers of his books is the MIX machine language, sort of as a handy reference as you're reading his texts, so that it's always there.

So anyway, to answer Luke's question, or to continue, probably the reason this is done, it's done because the boundary between the virtual machine and the compiler can be set by the language designers so that the virtual machine is very powerful or not very powerful. But as he suggests about cross-platform portability, and as we were talking about in the case of Forth, if you had something like a Java technology - or Perl, Perl's another great example because it's wildly cross-platform - if you implement the language against a custom-designed virtual machine, you can first of all design the virtual machine so that it's really good at doing the things the language needs, that is, it provides the facilities to, like, real-time garbage collection and memory management and very high-level constructs, much more than you would get from actual hardware. So that makes implementing the language on top of that virtual machine much easier and more convenient.

And all you have to do to move that whole blob that you've created, the whole implementation, to a different platform is rewrite a much smaller portion of it, that is, just the virtual machine for a different platform, and everything else that you've written runs. Because you've virtualized that the actual hardware into something which is not only non-specific for the platform, but also a close match to the things the language wants to do. So you can design a computer yourself that your language will run well on. And then you only have to implement that virtual computer that you've designed on the actual hardware.

Leo: So cool.

Steve: Yeah, it's just - look at all the cool technology we've come up with in computers over the years.

Leo: I know, I know. We're so lucky. Question 7, an anonymous listener worries

about the danger of computer microphones. We mentioned the danger of computer cameras. What about microphones? I'm worried, guys. There's been considerable amount of attention paid to hackers - or school administrators - using laptop cameras to spy on users. While egregious, this can be solved with a piece of opaque tape or a post-it note. But what about the built-in computer microphones, which are even more common than the cameras and don't even have a light that might let you know they're on?

Sure, images are embarrassing. But I could imagine audio could be just as bad, or worse, especially for identity theft. Yes, my credit card number? It's 2914.... How can I disable my computer's built-in microphone? Shy of opening the laptop case and disconnecting it, if I could find and get to it, the best idea I have is to delete the microphone's drivers. How helpful would this be in protecting my privacy? Thanks, guys, for an amazing show. What do you think?

Steve: I had one other interesting hack. Many of the laptops that have microphones built in also have on the side the little headphone and microphone jacks. They're normally colored, like, red and green - I think red is microphone and green is headphones - where you're able to plug those in. And the act of plugging them in disconnects the ones that are built into the machine. So that when you plug in your headphones, you're disconnecting the laptop speakers and routing the audio that's been amplified by the speaker amplifier out to your headphones. Similarly, when you plug in your microphone to the external microphone jack, you're disconnecting the microphone on the laptop and replacing it with the microphone on your headset. Now, I'm not sure that that's the case because aren't there instances, Leo, where someone, like when they're using Skype, they think they're using their headset, but they're actually still using the microphone on their laptop?

Leo: Happens to us all the time.

Steve: Is that when they're using USB headsets? Do you know?

Leo: Yeah, no no no. Let me think. So you're thinking of the interlock between the jack might be sufficient.

Steve: Yeah, well, what I was thinking was, if somebody wanted to easily disable the microphone in the laptop, just take one of the - some random piece of unused headphones, old headphones or something, and just snip off the connector so that you...

Leo: I think that would work.

Steve: I do, too. So you just have the little plug with no cable coming out of it, and stick it into your microphone jack. I mean, you could test it to see if it would work. And the idea would be you're just - there is an interlock which disconnects the laptop microphone when you plug in an external microphone, and it's an electrical interlock that nothing could bypass. And then so if you just snipped off the cord, so all you have is a little microphone plug, when it's plugged in, you ought to have no worries about your laptop

microphone being live anymore.

Leo: I think we want to test that. I think that's, you know, that would make sense, I mean, that it's a hardware interlock, and it just, boom. It's certainly not software. And you're right, it only is in the US- as I remember, it only happens when you have USB mics that somebody might be thinking they're on the USB mic. That's software.

Steve: Exactly. So, well, because you would see, for example, in your Skype, for example, it would say built-in microphone or USB headset. And so the user has forgotten to switch that over.

Leo: Right.

Steve: But if you've got...

Leo: Because that's a driver.

Steve: Right. That's a driver. So the idea being, but if you plug in a microphone into the little pink external microphone connector, I'd be pretty sure that it disconnects the microphone on the laptop.

Leo: A number of people are saying that on more modern machines like Macintoshes that's not the case. And we do know that the Macintosh has funny jacks.

Steve: Funny jacks.

Leo: Yeah, they're not pure analog jacks, at least the output isn't. And it may be the input is not also, that it may have some other thing going on. A number of people say in our chatroom it's all in software on the Mac.

Steve: Okay.

Leo: So if you couldn't do that, what would you do? You'd have to open up the machine, I think; right?

Steve: Yeah. And just deleting microphone drivers, it's not normally the case that you've got separate input and output drivers. Normally you don't have, like, separate drivers for the microphone versus the speaker. Normally you've got a single audio management DLL or driver which handles both directions. So you wouldn't want to disable the microphone because then you wouldn't have any speakers, which would seem like a bad thing.

Leo: That wouldn't be good.

Steve: Just turning the volume down doesn't work because software, that's all in software. So certainly any malicious software that was smart enough to turn your microphone on behind your back would know that it needs to use the mixer in order to turn the microphone input to the computer's A/D converter up and unmute it and so forth. So the only thing I could think of is, if you did have that physical microphone connection, see whether plugging something in there disconnects the laptop's microphone. If it does, you're in luck. Otherwise, I don't have a good answer.

Leo: Otherwise, you're screwed.

Steve: Just do sign language whenever you're in the presence of your laptop.

Leo: It's a really good question. We can so easily disable those cameras, but I don't - hmm.

Steve: Yeah.

Leo: It's a very interesting question. Well, certainly try the thing with a jack. Everybody's got something lying around. Your iPod headphones or, you know...

Steve: Well, I mean, yeah, exactly. All you need is the right size connector. Just stick it in and see if your microphone goes dead. Chances are it would.

Leo: I would think it would. Question 8, our last question. But there's two of them, so it's kind of like 8 and 8.1. Opher Banarie - who writes to the Giz Wiz all the time, by the way, Steve, you should know that. Opher is in Laguna Niguel, California, just up the road a piece. He says: He did WHAT? In SN-238 you discussed a question from Paul Welch in Australia about a caller claiming to be from a credit card company. He wanted to validate the caller, so he "asked for a number and called them back." What? No. Never ask the unknown caller for a number. The bad guys just give you a bad number to call.

You've got to call the number you already know to be legitimate, the one on the back of your credit card. I wouldn't even trust the number on the monthly statement because that could have been - whoa, this guy's paranoid - could have been intercepted in the mail and altered. I'm surprised you and Leo did not catch this authentication error and warn listeners not to call the number any unknown party provides. I'm a SpinRite owner with many boring success stories. Please keep up the great podcast.

And "Chuck," asking for anonymity, also writes about validating the caller. He says: Don't use my name. I work for a telephone company, and when I get a request from law enforcement for emergency tracing information, I always tell them I will call

them back. And then I open the phone book for that city or town and look up the police department. I then ask for the calling officer. And when the same voice answers, then and only then will I give him or her the information pending the subpoena. I never call the number they give me, only the printed phone number. This procedure has prevented me from giving out information more than once.

Steve: Well, I'm embarrassed not to have caught that myself. I should have. Of course I know better. I would also similarly react as our listeners did. It's like, whoa, don't ask him for the number and call him back. That's crazy. I mean...

Leo: Yeah, obviously.

Steve: ...you're not getting any authentication that way. So I'm embarrassed, and now we're on the record.

Leo: Yes. And now we're done with the show. Steve Gibson, you are the best. I really appreciate your doing this each and every week. Just great information that everybody can use. You can get more information like this - in fact, Steve really is a fountain of information - at his website, GRC.com. All 240 of our episodes are there in both the regular 64KB high-quality versions and 16KB versions for the bandwidth-impaired. You also have access to the transcripts there. Do you have transcripts for every show, all 240 shows?

Steve: Every show. I had Elaine - why can't I say her name? Elaine went back and did them all after we started doing transcripts. I said, let's get the rest of them done. So, yes.

Leo: That's so great. And I think a lot of times it helps to read while you're listening along.

Steve: Well, and it helps Google to find us, too.

Leo: Also very important. They can't search into audio yet.

Steve: Not yet.

Leo: And you'll also find, of course, SpinRite, the world's finest hard drive maintenance and recovery utility, which you just have to have. If you have a hard drive, you need to have SpinRite. And many free things like Wizmo, the one I just downloaded, which is such a - 38K. I thought, oh, it is zero bytes because it downloaded so fast. And then I looked at it, said no, it's running. It works. It must have gotten all 38 kilobytes. He writes this stuff in assembly language so it's light, tight, and bright. And, what was it, Dr. Dobb's...

Steve: I write it in read-only assembly language.

Leo: Read-only assembly. Dr. Dobb's Journal, the great computer magazine, used to have Dr. - it used to be called Dr. Dobb's Journal of Computer Orthodontia. And their tagline was "Running Light without Overbyte." And that was 30 years ago. Steve still lives by that slogan. Great stuff on the website. GRC.com.

Steve: And we'll remind our users, GRC.com/feedback is how people who listen can get their answers for their questions, just like the 8.5 that we read today.

Leo: Yay. Now, do you know yet what we're going to do next week, or is this...

Steve: Oh, yes. Hardware interrupts.

Leo: Oh, wow. Oh, wow.

Steve: Yup, we're ready.

Leo: So we're going to continue on the making of the machine.

Steve: Yup, building a computer, how all of the fundamental pieces fit and go together. We've established a good foundation now. We understand from last week about the notion of a stack and how things can be pushed and popped on it. And everything else we've learned provides us with enough foundation to talk about the evolution of how to do things lots at a time, how computers are able to manage hardware, which is very demanding, while doing a whole bunch of other stuff.

Leo: So you'll probably do polling as well as interrupts.

Steve: Yeah, got it in my notes already, Leo, exactly right. Polling is the first thing we'll talk about, and how and why that fails.

Leo: This is so great because partly it brings me back in time to the days when I was learning all of this stuff. And if you started in the beginning, as you did, and I almost did, it was much easier to get to understand all this because you kind of did it gradually, than if you had come to it now; right?

Steve: Yes, we were growing up with it.

Leo: We grew up with it.

Steve: Yeah. And in fact that's exactly the feeling that I've had when I think about, like, students now in computer science or technology, I mean, it's just like, oh, look at all this. It's just overwhelming.

Leo: Right, right.

Steve: But for us, we said, oh, look, here's a new language, and we sort of spent a week with it, and then we moved on. And so we saw them one at a time. Here, anyone starting out is like, look at all these languages. Where did they all come from?

Leo: You know what?

Steve: It's all just a big stew.

Leo: Steve, we're the last generation in history ever for a million years who will have been able to say we were in at the beginning of information technology. Interesting. Steve, I look forward to next week. Thanks for joining us. We'll see you next time on Security Now!.

Steve: Thanks, Leo.