Listener Feedback #110

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-286.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-286-lq.mp3

TOM MERRITT: This is Security Now! with Steve Gibson, Episode 286, recorded February 2, 2011: Your questions, Steve's answers, #110.

It's time for Security Now!, the show that helps you stay safe online. And joining us, as usual, the star of our show, Mr. Security, the man behind GRC.com, Steve Gibson. Welcome, Steve.

Steve Gibson: Hey, Tom. Great to be with you this week while Leo is off wherever he is.

TOM: That's right. If you were expecting to see Leo Laporte, I'm sorry for the shock.

Steve: Or hear Leo Laporte, I guess.

TOM: Or hear Leo Laporte, if you're on the audio version. Or read Leo Laporte, if you're reading the transcript. I don't know if I read different than Leo.

Steve: I don't think I took that into account, actually, back in the day when I was writing the code that creates the transcripts.

TOM: But we've got some good stuff to get to today. Some security flaws getting handled, I hope?

Steve: As always. And this is a Q&A episode, so we've got questions from our listeners, and comments, and feedback in general. So we'll go through that and have a good podcast.

TOM: Yeah, I'm looking forward to the Q&A. Haven't done a Q&A with you before. And as I was mentioning before the show, one of the parts of the show where you find the unexpected, when you get some of the most valuable tidbits, I think.

Steve: Oh, yeah. And our listeners, I have a feedback location, GRC.com/feedback, where we remind people they can drop their comments and questions and thoughts. And very often, in some cases, they're correcting something that I have had said, or asking a great question for an issue that I didn't highlight strongly enough in a prior podcast. So just lots of good stuff.

TOM: Yeah, I love that sort of hive mind that you can take advantage of with crowd sourcing, and people pointing out - and some people have more tact than others on it. But the end result is you learn a lot more because you take advantage of everybody's specialties.

Steve: Well, and when you've been out in front of an audience who are doing target practice for a while, your skin gets thick, and you don't take anything personally.

TOM: Right, you just pay attention to the information.

Steve: Just take the value where you can find it.

TOM: Exactly. I learned that in journalism school the first time I started doing reports and got ripped to shreds by the editor. I learned very quickly, don't listen to how he says it, just listen to what he's saying, take that, and learn from it.

Steve: Exactly. Just say, "Thank you very much for your input."

TOM: But we always try to say things in a very nice and acceptable manner.

Steve: Why not? Well, we've got a couple updates, security-wise, and a bunch of news. The big news since last week is that a new flaw has been found in IE. Microsoft has about five other problems that they've been tracking since the beginning of the year, which we ought to be fixing hopefully before long. This one is something that most users of Windows won't need to be concerned with, if they're not using IE. It's in the so-called MHTML module of Windows, which actually is not a formal part of Internet Explorer. This is that feature which only IE has, which allows you to save an entire web page with all of the other assets of the page as a single file. It creates that HTM, is that what - no, I'm sorry, the MHT file is the so-called "IE archive."

And what someone discovered, and there is proof-of-concept code now in the public, and exploits are expected to be happening before long, someone discovered that there was a mistake in the way this MHTML module was rendering pages that were stored in this format when they were redisplayed, such that you could exploit scripting in this stored archive in order to get arbitrary malicious code to execute. So Microsoft is working on a fix. But first of all, anyone using another browser - I'm not sure about Opera, though, because Opera also supports this compressed format. But I know that Firefox and Chrome - Firefox requires a plug-in and has a different format; Chrome doesn't support it at all. Safari doesn't support it at all. So other non-IE users are almost certainly safe.

Microsoft does have one of their little Fixit buttons which can be deployed. And frankly, if anyone's concerned about this at all, it's easy to use this Fixit, and there's almost no side effects, meaning that there isn't really a downside. What you're doing is you're turning off scripting support in these MHT files, these IE archives. And it's hard to imagine that when you want to save the whole page with all the stuff in it, it's hard to imagine that disabling scripting whenever it is you get around to looking at it again would be much of a problem.

TOM: So this means that, when you save one of those pages as an archive, it won't execute it when you view it later?

Steve: Correct. It won't execute the script, any JavaScript that you saved along with that. I think...

TOM: So it doesn't affect the saving, it affects the viewing.

Steve: Correct. And in IE it's that option that IE has always had. They added this in IE5, which is "save web page complete" I think is what it actually says in the dropdown box. I've used it in the past. It's kind of handy to save - if you've got a page, you really want to keep it, not just the HTML, but all the other pieces of it, it bundles them all together. In fact, MHTML stands for MIME HTML. MIME is the multipart extension of email, where you're able with MIME to essentially attach different types of objects, binaries, and pictures and code and so forth into an email.

So this is MIME HTML, which is a non-standard format. This is not part of any larger web standard. This is just IE which has created this. What users can do is go to support.microsoft.com/kb/2501696. So again, that's support.microsoft.com/kb/2501696. You'll find there one of those little one-click Fixit buttons that Microsoft is doing now, which what it does is it adds some lines to the registry to disable scripting for files with this MHT file extension. That is, it disables the script handlers, JavaScript, for those documents. And it does nothing else. So it's an innocuous, worthwhile fix, I would say, if you're a person who is more likely to encounter malicious stuff; if your habit is to be browsing around out on the Internet, maybe in some shady areas.

It's not clear how soon Microsoft's going to fix this, when that's going to happen. We are, what, we're February 2nd, so we're not yet at the second Tuesday of February. So we don't have perhaps too long to wait. Of course yesterday, Tuesday, was February 1st. So the second Tuesday of February will be the 8th, which is as soon as it could possible come. Maybe Microsoft will have it fixed by then, although that's a short window. This just happened in this last week. So this may not be time for them to fix it. We may be waiting till March. Or they have said they may push out an out-of-cycle update if this thing goes exploit-crazy in the wild. So the way it's exploited is there is a way that a user can simply visit a website which uses cross-site scripting in order to exploit this MHT problem without a user having to store and view something. So it's not as if it's necessary for you to store a bad page and then play it back and get zapped.

TOM: It's not side-loading an MHTML document itself, it's just exploiting the functionality?

Steve: Well, the page that you go to and load can say to your browser, load this little snippet of MHT file, and then use the JavaScript in order to execute that and exploit it. So it is still a one-click deal. It's where you visit something malicious on the web, and you're toast. So my feeling is, if you're concerned - first of all, only if you're using IE bother with this because, if you're already on Firefox or Chrome, or probably Opera, then for sure you have nothing to worry about. And Microsoft, I'm sure, I mean, there is exploit code out. They're aware of it. They're working on it. So with any luck this thing will be fixed pretty quickly.

TOM: And if you're running NoScript, presumably that protects you, as well; right?

Steve: Exactly, because it does require scripting in order to invoke this problem. But if you're running NoScript, then you're on Firefox, so you're safe anyway.

TOM: Exactly, it's sort of a catch-22 there. I use Firefox to save web pages complete anyway because it just saves them as HTML and then puts all the assets in a folder, which I find is much more convenient for being able to look at it again in whatever browser I'm using. Whereas the MHTML is not handled by all browsers.

Steve: Right. So what Firefox does is it saves the page, and it converts the web links into relative folder references. So it pulls it out of a folder located right underneath where it's saved the page. And so you're able to, as you said, redisplay the page not having to have an Internet connection.

TOM: It's a great way to cache pages, as well. Speaking of Opera, Opera now has a new version.

Steve: Yes. They had 11, actually hasn't been out for long, when a problem was found that caused them to do 11.01, which is where they are now. Opera doesn't have a huge market share, but I know that we've got listeners who really like Opera. They've gotten used to it. They like the features that are unique to Opera. So I wanted to advise them that, first of all, that 11.01 has happened. So make sure that that's what they're running. That one fixes a bunch of cosmetic UI problems, a handful of minor security issues, and one worrisome remote code execution vulnerability in Microsoft's handling of the "Select" element in HTML forms. If you loaded a web page that had really long input in the forms, and you could use script in order to load the input, which is what a malicious page would do, that there was an integer truncation problem that would get overwritten. And we know the rest. Then the bad guys got control of your machine.

The problem is that Security Focus is also reporting another integer overflow problem in the option, the form "Option" element that still exists in this latest 11.01 release. It's not clear that there's any active exploitation of it. Opera does not have a patch. But they are working on it. And with any luck they'll have a .02 release that'll fix that one. But just so I wanted to give Opera users a heads-up, first of all, that Opera had recently gone to .01, that is, v11.01; and that there is still - and there's a known problem with this one, but I imagine Opera will be getting to it and fixing it quickly. So, because it's very related to the other one.

TOM: So there are your security updates. Let's move into the security news. Google today had a big announcement about new features in Honeycomb. But we've got some issues with Android's v2.3, Gingerbread, which a lot of people are very excited to get on their phone. If you're running the Nexus S, you actually have it on your phone right now. What's going on with this?

Steve: Right. What happened is that there was a known problem with the previous release of Android 2.2 and a data disclosure vulnerability, as it was called, which allowed a malicious link, essentially, either in email or browsing. So the email link would take you to a web page that was configured to exploit a problem in the browser that would give this malicious site access to your device's SD memory card. And that might contain financial information, photos, banking records...

TOM: Anything you store on there, yeah.

Steve: Yes. Essentially they had access to all the files on the SD memory card. So Google fixed it and said, ta-da, here's Gingerbread v2.3. Well, an associate professor at North Carolina State University recently informed Google that he had found a way around their fix in this most recently released v2.3. So essentially this thing has the same problem. Google is aware of it. They have verified it, and they're moving to fix it. The

only thing users can do is you could disable JavaScript. That's one course of action. Because as, I mean, all of our listeners, Tom, are used to my saying, well, if you disable JavaScript, that solves the problem because of course scripting is just such a problem from a security standpoint. So here, even over on Android, disabling JavaScript, or using a different web browser. Instead of using the default web browser, use one of the alternatives that doesn't have the problem. I'm sure Google will get this fixed and pushed out quickly, but there is a known vulnerability in Gingerbread that they thought had been fixed. But turns out there's a way to work around that.

TOM: As with anytime you're using JavaScript, be extra careful about where you're clicking.

Steve: Exactly. And SourceForge, the major open source archive and project management site, was hacked.

TOM: Oh, yeah. I heard about it. This is horrible because SourceForge, if people don't know, is a great place to find open source software. I mean, if you're ever looking for a free version of anything, if you have an idea of a utility that you wanted, go search SourceForge because chances are, like Google Code, one of those two places has somebody who's been working on something. But you may be a member of SourceForge if you're a developer, if you're uploading things. And your data may have gotten hacked here.

Steve: Well, yes, and that's the question. There are 230,000 open source projects that are being managed over at SourceForge. And what happened was they discovered that somehow a malicious SSH, a secure shell daemon, had gotten into their servers, and it had been maliciously modified to perform password capture of anyone logging into SourceForge over SSH. So they immediately blocked all logins, and they just shut down login because they could no longer trust the integrity. They sent email to everyone who had a password, telling them to please change their passwords, after expunging this known malicious SSH daemon from the servers.

And now they're in the process of going back through and performing some sort of audit of the code base because, I mean, as you said, a huge number of high-integrity, highly used open source projects are being managed there. And the question is, and many people have posed the question, did the bad guys use the password capture to give them access to any major open source projects and modify the source code maliciously so that what was once good source code is now - it's got some hidden backdoors in it. And that's the question that these guys need to answer.

So we don't really know how bad it is. I'm hoping that they're actually able to perform an audit. Bad guys have a way of, especially bad guys who know what they're doing, of covering up their own tracks and, for example, changing modification dates and erasing modification logs and so forth in order to not be found. So it is scary for a big major site like this to have found themselves with a backdoor that was in active use at the time that they encountered it.

TOM: Now, can they crowd source the analysis, the code audit, and ask people to kind of pitch in and look for things? Or is that untrustworthy in this case?

Steve: Yeah, that's the problem. The only way I could see they could do it, they could run maybe a source compare versus offline backups, see whether anything has changed that doesn't look like it's changed. And that would allow them to find it because they probably have offline backups that would inherently have been unavailable to a bad guy. So if they were to do a mass comparison of files that showed that they had not been

changed, to actually see if they had been changed, then I would think that's partly, probably, one of the things that they're doing.

TOM: Quite a mess. I mean, you can still go and download source code from SourceForge. You can download things from the accounts. But is that safe?

Steve: Yeah, that's the question.

TOM: I mean, you're always taking a risk when you download anything on the Internet, of course. But you've got a little extra risk to frost the cupcake, in this case.

Steve: Well, yes. And, for example, we heard, our listeners heard, there was that rumor toward the end of last year that OpenBSD's crypto library had a backdoor installed in it a decade before by the FBI. And people ran around and really scrutinized the OpenBSD source code. That rumor was debunked. It was found that there was nothing bad that happened. And in fact they did find, by looking at it so closely, they found some things that they wanted to fix because this is code that hadn't been looked at for 10 years. So they said, oh, well, while we're here we'll fix these other things.

So people feel very good about source being open because it allows scrutiny. But of course it also requires scrutiny, or you're not getting any value from the source code being open. So I will keep my eyes out for any further news about this. And I hope that maybe the bad guys were discovered, this backdoor with the SSH daemon was discovered before anyone had had a chance to do anything bad. Or that the passwords which were captured were to minor projects. Hopefully they know which passwords were captured, and then they would know where to look. That would hugely narrow down their search.

TOM: That would make things a lot simpler.

Steve: Yeah.

TOM: All right. We are scraping the bottom of the barrel of IPv4 bucket. Well, wait. We're scraping the bottom of the IPv4 barrel. It's not a barrel and a bucket at the same time. In any case, the last two freely assignable /8s were given to App Nic. And now they're down to five, which triggers an interesting rule; right?

Steve: Well, I wanted to just - alarm bells have been going off with people because last week I told everybody about the Twitter user which is @IPv4Countdown. And shortly after bringing that to everyone's attention, the countdown really seems to have accelerated. I mean, it just started going. It was like almost every six hours they would be announcing that another /16 had disappeared, and they were down to - I think we were at 32 million IPs, and now it's like it's almost gone down to nothing. And so I wanted to assure people that there's a little bit of a misnomer to this. This is the Internet-Assigned Numbers Association, the IANA, handing out, as you just said, Tom, to the major registries the final remaining big blocks of otherwise not previously allocated space.

TOM: The five families of the Internet.

Steve: Right. This we knew was going to happen early in 2011. This is different than what is expected to happen late in 2011, which is actual exhaustion. So the idea is that, out of the 256 possible, zero to 255, first byte of the IP address, there had been a big chunk, well, not a big chunk, but a sizeable number that had never, ever been given out.

And so early this year, and this is what this countdown has been watching, is the IANA finally essentially relenting and handing out the remaining large, never-before-allocated blocks of the Internet space to these five registries. Then they, during the next six or seven or so months, will be handing these newly received blocks they just got from the IANA, they'll begin doling out chunks, I mean, jealously and carefully and with minimal waste, doling out just what IP blocks they have to. They're going to want to conserve this remaining space.

So, I mean, for several years now there's been back pressure on people who wanted IP blocks. I remember when I first got on the 'Net a decade ago, you could just say, oh, yeah, give me 256. I want a /24 network. You could get large allocations with no problem. When I moved GRC over to Level 3 about three years ago, I had to fill out what they called an "IP Justification Form." And I was begging for the 16 IPs that I have. Whereas on my two T1s, I've got 64 sitting here. I mean, me, where we just connected with Skype.

TOM: You're on a wealth of riches.

Steve: And it's funny, too, because when I switched over to Cogent from Verio, I said to the two tech guys who had gone from Verio to Cogent and so who knew me really well, I said, guys, I really don't need all these. And they said, well, you've had them before. You might as well just still have them. It's like, okay. So they were being very free and loose with them at the time. That is certainly not the case anymore. Now you've got to beg and plead for what few IPs you're able to get from an ISP. Generally, as long as you're using them, you can keep them. But there still are all kinds of people, I mean, major corporations and networks, that are hoarding their IPs and don't want to let them go. And of course the pressure is increasing on them giving them up.

So I did want to make sure people understood that what this IPv4 depletion was showing was really only the obviously visible depletion that's being monitored where the IANA is saying, okay, we've peeled off the last of these, handed them out to the registries. Now they will individually hand those out over the next six or seven or eight months. And I've said on this podcast many times in the last few, actually few weeks, that we'll be watching this IPv6 drama unfolding. And in fact we've got a couple interesting questions from our listeners about that, too.

TOM: Yeah, June 8th will be IPv6 Day, when several big websites are going to test out their IPv6 capability. But the interesting thing is, even after that day, they're switching it back off.

Steve: Exactly. And in fact we talked about that a couple weeks ago, saying, okay, so the Internet is running out of IPv4 IPs, like maybe in October or November? And they're waiting until summer of this year to do, like, the big test? It's like, okay. So, yes. We're in for some very interesting times.

TOM: If you go to IANA.org you can find the page that has the address space registry and what blocks are allocated and what aren't. There's only five listed as unallocated right now. Those are the ones Steve has mentioned are going to be given to the five different regional Internet registries, the RIRs. And then once those are there, different ones have different policies. But APP NIC, the Asia Pacific one, says that they think they've got about three to six months before they get down to the very end, and then they're just only going to use what else they haven't allocated as sort of transitional material to try to get people by until they move to IPv6.

We were talking to Dane from Sonic.net last week. He thinks we'll never get off IPv4. He

thinks we're just going to continue to dual use. And it will slowly trend towards being predominantly IPv6. He doesn't think we're going to have a place where suddenly the Internet breaks. But he also thinks we're never going to get off the crack. There's still going to be IPv4 addresses used out there.

Steve: Well, actually I'm of exactly that opinion. Consider that there are four billion of them. And while, yes, there's a problem, for example, with all the cell phone usage, where every cell phone needs an IP address. IPv4 has a subset on an IPv6 address. So if you have, I think it's all zeroes, and then two blocks of FFs, and then an IPv4 address tucked in at sort of the least significant bits, the least significant 32 bits of the IPv6 128-bit address. So there is a formal spec for the way the entire IPv4 address space can always, for all time, it will always live in a small corner of the formal IPv6 address space. So you could either use IPv4 to get to GRC, for example, I'm 4.79.142.200, so you could use that. Or you could use this 0000.FFFF.FFFF and then the same thing, 4.79.142.200, which is an IPv6 address that will take you to the same place.

So I'm in complete agreement. I don't see a day ever, ever, where IPv4 stops being routed. It'll always exist. We'll have it in our machines. And the sad thing is that this is going to be a mess. We've got some questions in today's Q&A that sort of highlight just what a problem we're probably going to be dealing with. So, I mean, it's going to be great topic fodder for this podcast as we move through, wade our way through this whole IPv6 and v4 conversion.

TOM: Do you think anybody's going to come knocking at your door, asking you to spare an IPv4 address? Would you be able to hand some over, if they did?

Steve: Absolutely. And I would be glad to do so. If Cogent ever says to me, hey, Steve, we were once really generous, but how many do you really need, I could get by with just a very few here. I mean, I'm using all the ones that I've got at Level 3. I don't need more, but I couldn't survive with fewer. But, yeah, I'd be happy to give Cogent back a block.

TOM: I think that'll happen in a lot of corners, and it'll extend the life of IPv4 a lot longer than maybe it looks on paper right now because you're not the only one who received them in that time when people thought, eh, there's plenty, take a bunch.

Steve: Right. And probably what'll happen is, I mean, it would be obvious to an ISP who was monitoring traffic, for example, that I'm using three or four IPs out of my block of 64. I mean, they would just simply see no traffic over those addresses through some length of time. And I would imagine, if they then - so it's very easy for them to see IP addresses that are not being used. That would allow them then to come back to me and say, hey, you don't seem to be using these. Are we wrong? Or do you just have 54 computers that you haven't turned on or so forth. Then I'd say no, you're not wrong. I don't need them. And they'd say, well, we want them back. And I'd say, fine.

TOM: Wait till the IPv6 crunch comes, once the...

Steve: Well, it'll be really interesting to see what happens. But I really think you're right. And of course we've got NAT, and we'll be talking about that in our Q&A today.

TOM: All right, let's move on to Google and Connecticut. Connecticut got a new attorney general. The old attorney general was really after Google over this WiFi slurping. The new attorney general has settled.

Steve: Well, yeah. I mentioned this once before. We've sort of been following the whole Google WiFi-sniffing mistake on the podcast from its very inception. And I did mention a couple weeks ago that the previous Connecticut AG was demanding that Google hand over all the data that they had captured during their inadvertent sniffing of WiFi in Connecticut. Google refused, and so it looked like they were going to roll up their sleeves and go to war. What happened was Google settled, essentially, without needing to go to court, with the formal acknowledgment that Google had inadvertently collected information, including partial and complete email and addresses of requested web pages.

And so essentially Google formally acknowledged what anyone looking at the raw data would have concluded. I mean, this is entirely reasonable that Connecticut would be satisfied with Google's disclosure and acknowledgment of that, rather than actually requiring the disclosure of the raw data itself. I mean, it made no sense to me at all when Connecticut was saying this is what they wanted. And so I was glad to see that Google said no, and then was able to get agreement with Connecticut. Who knows whether Connecticut's going to go any further with this. I'm hoping this whole thing blows over because it was clearly just a configuration mistake on their data collection side. And so much more has been made of this than needed to be made.

TOM: More privacy violations have happened with governments insisting on looking at the data than would have happened if Google had just deleted it immediately.

Steve: Right, right. So in what I would have to call the "looney tunes" announcement of the week - which many of our listeners sent to me. I got a bunch of Twitter input from the people who are following me on Twitter. Computerworld reported a story from the Intel CTO, the chief technology officer of Intel, saying that they, Intel, have new technology on the burner, coming along soon, that will stop zero-day attacks in their tracks.

TOM: From the chip side.

Steve: Yes, some sort. It's not clear that it's only hardware. And they did say that this was underway prior to Intel's acquisition of McAfee. So this is not something that they got from McAfee. They're claiming that some new feature of their chips will prevent zero-day attacks.

TOM: Steve? I hate to say it, but this almost sounds too good to be true.

Steve: Gee, you think, Tom? Okay. I call it "looney tunes" because, okay, a zero-day attack, as our listeners know, is nothing but a vulnerability which is first discovered because it's found in the wild, being exploited, rather than found by a researcher and divulged to the people who are being potentially exploited, or who are maintaining the whatever it is, the software that is vulnerable, that allows them to patch it before the problem is known. So...

TOM: You've got zero days to prepare for it, in other words.

Steve: Precisely. Everybody learns about it when they see it actually being exploited. So I have to share with our listeners what Computerworld wrote because it quotes this guy from Intel. So the article, January 26 Computerworld, says: "Intel's chief technology officer says the chip maker is developing a technology that will be a security game changer. Justin Rattner told Computerworld on Tuesday that scientists at Intel are working on security technology that will stop all zero-day attacks. And, while he would give few details about it, he said he hopes the new technology will be ready to be

released this year.

"'I think we have some real breakthrough ideas about changing the game in terms of malware,' Rattner said." Continuing the quote, "'We're going to see a quantum jump in the ability of future devices, be they PCs or phones or tablets or smart TVs, to defend themselves against attacks.'

"He noted that the technology won't be signature-based, like so much security is today. Signature-based malware detection is based on searching for known patterns within malicious code. The problem, though, is that zero-day, or brand new, malware attacks are often successful because they have no known signatures to guard against."

Still reading from this Computerworld article, "Intel is working around this problem by not depending on signatures. And the technology will be hardware based, though it's still unclear if it will have a software component."

And then Rattner again was quoted: "'Right now, anti-malware depends on signatures, so if you haven't seen the attack before, it goes right past you unnoticed,' said Rattner, who called the technology 'radically different.' 'We've found a new approach that stops the most virulent attacks. It will stop zero-day scenarios. Even if we've never seen it, we can stop it dead in its tracks,' he said."

TOM: So what it sounds like to me, if I had to read the tea leaves, is that they've figured out a way to prevent against some attacks based on not having a signature. And that's not entirely new. People have claimed that before. But the idea of coming out and saying they've solved zero-day vulnerabilities, that's just grabbing for press attention. They can't have done that.

Steve: Well, yes. What they're doing is they're saying, if we take it literally, they're saying we're preventing there ever from being an attack which is not known. Which, I mean, is nutty. And, for example, we've talked about the so-called "execution disable flag" which exists now in all Intel architectures, where, for example, you can set this flag for the pages of memory which the stack lives in, and the pages of memory which contain data. And the Intel chipset will refuse to execute code from those memory pages that have that bit set. So there we have hardware which is enforcing and preventing stack overflow execution problems. Yet we still have those problems.

I mean, so, yes, it's better than not having it. But it didn't, like, immediately solve buffer overrun problems. We've got the bits. Everyone's using them now. And we still have buffer overrun problems. So, I mean, on one hand you're sort of tempted to go, wow, Intel, how could they be wrong? On the other hand, how could they be right?

TOM: It's not like Intel is like Steorn, the perpetual motion machine company from Ireland, just making some ridiculously crazy claim that no one will ever believe. It's Intel. There's something behind this. They've just - they've notched the rhetoric up a couple times, as indicated by using the word "quantum" to mean some sort of amazing advance. It's just all puffery. I'm sure that they've got something that is pretty interesting, and hopefully will advance security. But you know more than anyone else it's an arms race. Nothing is ever going to prevent everything.

Steve: And frankly, I mean, for a company as big as Intel, I'm sure the CTO means well. But he's a C-letter executive who probably heard something interesting from the lab and doesn't really understand it himself. So anyway...

TOM: Yeah, there's a difference between, "Yes, sir, this will stop zero-day vulnerabilities," and "This will stop all zero-day vulnerabilities."

Steve: Or we've come up with an interesting idea that may help us to improve the security of future chipsets when operating systems and all programmers behave correctly.

TOM: Well, because you're not going to fix the user. I mean, even forgetting everything else, the biggest vulnerability is the person using the machine.

Steve: Yeah. A lot of really smart people have been looking at this problem for a long time. And it still escapes us, that is, the solution to it still escapes us. So it would be nice to have maybe some sort of hardware support that will give us a better handle. Claiming that it stops zero-day attacks is just nutty, and it makes for great press.

TOM: Yeah, exactly, makes for good headlines. Call me when they have a patch for PEBCAK. All right, let's move on to - actually this is a little bit of a patch for PEBCAK - Facebook finally implementing the ability to have a secure connection, have an SSL connection all the time.

Steve: Well, I announced it last week, yes. I announced it in our podcast last week. That morning, literally just a few hours before the podcast, Facebook's blog, their security blog, announced that they would allow users to optionally set the enforcement of SSL secure connections whenever they're using Facebook, whenever possible. However, at that time my own Facebook account, which I use for testing purposes, did not yet show that checkbox. We saw a screenshot of it in the blog posting, so I knew what it was going to look like. And I did get some Twitter feedback saying, "Hey, Steve, I saw your tweet about this, but I don't have it yet."

So I did want to let everyone know that now, a week later, I know you said you found it. I have found it on my Facebook page. So, although I did, just in the chatroom chat before we began recording this podcast, someone mentioned that their dad's Facebook page still didn't have it. I consider that anecdotal. I bet it's there now, as long as they go look for it in the right place.

So I did want to follow up and just say, if any listeners excitedly went to their accounts after hearing about this announcement last week and were disappointed not to find it, check again if you haven't. I'll bet it's there now. Turn that on. And of course what that does is, it means that you have an SSL connection for all of your use of Facebook.

I did see that some third-party applications running on Facebook may have a problem with this, and may not support it, which may be why it's not turned on by default. So if you turn it on, and things break or don't work, first of all, I'd love to know about it: GRC.com/feedback. But by all means, I immediately turned mine on, and it just means that when you're roaming around, especially in unsecured open WiFi hotspots, that you're not going to be subjected to abuses by Firesheep and its ilk.

TOM: Yeah, go to Account, and then you want to get the Account Settings. And on that main settings page, Account Security, you should have it say "Set up secure browsing (HTTPS) and log-in alerts." And then you click Change for that, and then you're able to hit a checkbox that'll allow you to do it. If you don't see that, you haven't got it yet.

Steve: And do make sure you click Save, which is right underneath that checkbox.

TOM: Yes, absolutely.

Steve: Because just turning the checkbox on doesn't do it. You've got to save that change.

TOM: Very good point. All right.

Steve: I have some feedback from a listener about SpinRite. His email handle is LeeBing, but his name is Leland. And he sent an email actually to my tech support guy, who forwarded it to me, saying "GRC Thanks." He said: "Please let Steve know that it may not have been a miraculous story, but still extremely thankful for SpinRite. My wife runs her own business from home. Internet Explorer had been acting odd, some crashes lately, so she didn't think much of it when it crashed the other morning. She closed out and rebooted her computer like normal, except this time it wouldn't boot, not even into safe mode. It would try to start, but just automatically reboot again on each attempt.

"Not wanting to have to reinstall all her software applications and settings if I didn't have to, I spent over three hours getting a new install of Windows XP into a different partition, hoping to then run a virus scan on the offending partition, assuming that that had caused the problem. But I couldn't read the drive at all.

"After many other attempted workarounds, I knew I had one good option remaining. I knew from all the stories shared on Security Now! that there would come a day that I wished I had a copy of SpinRite on hand. I downloaded and purchased SpinRite from another computer. Oh, I hate to think if I hadn't had a second computer working in the house.

"This was about 10:30 p.m. on a Wednesday night by this time. Within five minutes of starting, SpinRite found a bad sector and started recovering the data from it. Within 15 minutes it was done with that entire partition. Upon rebooting, I waited while Windows corrected some cross-linked files and such. But then it came right up into my wife's normal screen, and the machine has been running fine since then."

Then he has this in caps, he says, "PLEASE PASS ON TO OTHER LISTENERS! Don't wait until the day comes that you wish you had SpinRite on hand. The day you...." And he's saying this, I'm not. "The day you need it, you may not be able to simply walk over to another computer and download it." Well, you probably have some friends who could help you out. Anyway, he says, "And stop wasting hours avoiding using it or delay getting it. Had I started with SpinRite when I got home, I would have been in bed early instead of up late. Thanks again for the great software and great work on Security Now! and GRC.com. Sincerely, Leland in Raytown, Missouri."

TOM: It's a good point. Have your tools downloaded and on a CD or USB drive. It'll save you time.

Steve: Well, it does. I don't realistically expect people to buy SpinRite when they don't need it. But it certainly, I mean, what we do see is that having SpinRite on hand and, more importantly, running it preemptively, solves this kind of problem. Leland never had a chance to buy it. Had he owned it and run it, it would have fixed this problem prior to it getting to the point where that machine would no longer boot. So really I think that's the great benefit.

TOM: Steve, it's time for another listener feedback episode, Episode 110 of Listener Feedback. We've got nine questions today. First one comes from Ben in Atlanta, who

doesn't quite understand whether XORing for encryption is good or bad. He says: I'm confused about your opinion on using XOR in encryption. I believe you previously said that it is "bad," but in the episode on Bluetooth you spoke briefly about RC4, and you said that it was fine to XOR with the RC4 stream. Could you please explain the issue?

Steve: Okay. So this is confusing. I guess the way to explain it from a thousand feet is to say that XORing itself is a useful and valuable technique for mixing a pseudorandom stream with plaintext in order to convert into ciphertext. The idea being that, if you - and we did cover this at length in our crypto episodes. So a listener who wants, like, more in-depth coverage of the topic can certainly go back and get a whole podcast on how this works.

The idea is that the XOR operation is essentially a conditional bit inversion. If you XOR zero and zero, you get zero. If you XOR zero and one, you get one. If you XOR one and zero, you get one. And if you XOR one and one, you get zero. So if you were to, like, write that down on a napkin and look at it, you would see that, essentially, if you consider one of those bits of input to be the plaintext, and the other bit of input to be like a control bit, that control bit, whether the control bit is one or zero, determines whether the data bit is inverted or not. So if the control bit is zero, and you XOR anything with a zero, you get the same thing. If you XOR anything with a one, it inverts the data.

So what's so cool is, if you have a source of pseudorandom data, pseudorandom noise, then, almost counterintuitive though it is, if you take regular plaintext, and you XOR it with the pseudorandom noise, the pseudorandom noise being sort of that control bit, it determines whether the bits are flipped in the plaintext in a random, a pseudorandom fashion, which performs as good an encryption as exists. That is, it's absolutely unbreakable, given some limitations.

And so I think this is where Ben is confused and where I have talked about XORing and the problems with XORing. Given that you absolutely never reuse that pseudorandom data, and that's crucial, and, well, with that single caveat that you never reuse that pseudorandom data, so that it's always different each time you perform an encryption against plaintext, then you've got absolutely bulletproof encryption. The beauty of it is that, when you take that encrypted data and so the same process again, XOR it again with the same pseudorandom data, since you'll be reinverting the same bits in the encrypted data, which you inverted to create the encrypted data, it reverts it to plaintext.

So it's of tremendous value, for example, in wireless communications. We use it right now in WPA, which is an industrial-strength, bulletproof encryption technology which works great. It does require that it be handled properly. And it is the encryption that's used in Bluetooth. Bluetooth uses a pseudorandom bitstream and XORs the plaintext to create the ciphertext. The weakness comes from, if you know what some of the encrypted text actually is in plaintext, the so-called "known plaintext approach," if you know what that is, then you can XOR what you know with the ciphertext and recover the pseudorandom bitstream.

So that allows you - essentially we have three things. We've got the bitstream, the plaintext, and the resulting ciphertext. Any two of them will give you the third. So if you know what the plaintext is for corresponding ciphertext, you can XOR those, and what falls out is the pseudorandom bitstream.

TOM: And that's why you don't want to reuse it, right, because then it could be used to unlock anything else you've used with that stream.

Steve: Exactly. Exactly. So if you ever made the mistake of reusing that same stream,

now that you've been able to recover it, knowing some of the data that was encrypted, you can recover other data which you don't know the value of. And so the whole thing falls apart. So while it works, and it's very useful, especially in any situation where you just don't have much computing power, you have to be very careful with the way it's applied. So I like it. But, again, people have continued, over time, people who have used XORing for encryption have made mistakes that ended up biting them. And so essentially the world sort of moved away from it, over to, for example, AES encryption, which is far stronger and doesn't have these same sorts of weaknesses. So, yes, I like it. But you just have to be very careful with the way you use it.

TOM: Right. And you don't have to be as careful with AES.

Steve: You have different - you have to be careful in different ways with AES.

TOM: It's not as easy to mess up?

Steve: Correct.

TOM: Yeah. Question #2 comes from Chris Lincoln in Fremont, California, who thinks he's found another mass cookie OptOut solution. I think you mentioned before about Firefox and Chrome having their own plans for opting out. Chris says: Steve, on the recent Security Now! podcast you mentioned the script on aboutads.info didn't get you many options to opt out from online ads. For Firefox users, I recommend the "Beef Taco" add-on. TACO stands for "Targeted Advertising Cookie Opt-out." Cookies and tacos, and it's lunchtime. I'm getting hungry. It registers local cookies with the opt-out setting for over 100 online advertisers, and the cookies stay even when clearing the cookies within the browser. This makes reading Spybot S&D scans tedious, but it eliminates the majority of what came up in my scans anyway.

A lightweight TACO, known as "Beef Taco," is available at GitHub and Mozilla Add-Ons and sets 132 opt-out cookies. It is lightweight, completely non-intrusive, and you can see the full list of cookies at github.com/jmhobbs/beef-taco. Note that it includes Google/DoubleClick. I'll take a look. Did you take a look at this?

Steve: I did, and I wasn't clear what I was seeing. There were some complaints about v3 having become bloated and a real problem, and people saying just stick with v2. Essentially, we talked last week about this aboutads.info page. And what's there is a script which allows advertisers who are participating to sort of log themselves in, log in their participation with this website. And then you run a script there, JavaScript, which causes your browser to go to each of those advertisers, asking to please receive an opt-out cookie from them. So my complaint was that, while there were, like, I don't know, 60 or 70 apparently present advertisers, when I tried that aboutads.info page, I was only able to opt out of maybe nine or 10. It said that it was unable to get the other ones to behave. Maybe they've registered, but they don't yet have handlers for this aboutads.info page. So it didn't seem very effective.

So I just wanted to bring Chris's mention of Beef TACO to our listeners' attention. What this is, is different. This of course is - it's something which is sort of pre-installing those opt-out cookies for you. You don't have to visit those websites. It just comes along with 132 sort of premade cookies, which it will just automatically put into Firefox's database when you run it. So it's sort of like a built-in blacklist for opt-out advertisers. And the reason he mentions that Google/DoubleClick is also listed there among those is that I did note that Chrome was promoting a solution also, which interestingly did not include their own advertiser, DoubleClick. And so he was just bringing up the point this week that, well, this solution does include that.

So this is sort of another approach. I wanted to let our listeners know about it. I'm sort of annoyed at the idea of installing 132 cookies into my browser. I'm using NoScript and AdBlocker and so forth to not go to these places. But I could see that this could appeal to some people.

TOM: Yeah, I think NoScript and AdBlocker are probably your best bet, if you're really concerned about this. But the FTC has called for some sort of opt-out list, and we're going to see more of these kinds of solutions being put out there.

Steve: Right.

TOM: Question #3 comes from Lucas J. in Maryland, wondering about delaying IPv4 depletion. You had said we were going to get back to this topic earlier in the show. Lucas asks: Could ISPs, phone companies, et cetera, help delay IPv4 depletion through NAT (Network Address Translation)? I know I've heard before of ISPs putting their whole customer base behind one large NAT, giving them all one Internet-facing IP address. Couldn't we do this on a large scale? All Comcast customers on the East Coast could have one of a handful of IP addresses; all Verizon phones have one of a handful of IP addresses. Wouldn't this help, or are there more downsides than positive benefits?

Steve: Well, yes. First of all, it is NAT which has allowed the Internet to survive as long as it has. I mean, I know that all of our listeners have NAT routers, and these days probably have many more than one IP operating in their own home networks. So all of those IPs, all those machines behind their home NAT routers are seen publicly as a single IPv4 address. Now, it is the case that some ISPs have for years been running their users on, for example, 10-dot addresses. The 10-dot network has 16 million IPs behind it. So it's all the IPs beginning with 10. And then you've got three more bytes. So that's 24 bits of space that is 16 million different combinations.

So an ISP that has up to 16 million customers could put the entire customer base on a nonroutable 10-dot network behind NAT. You probably, well, you almost certainly cannot get by with one Internet-facing IP address because the way network address translation works is it essentially uses port numbers. Technically it's called Network Port Address Translation, although people shortened it to NAT. It actually uses port numbers to disambiguate the target IP for traffic coming back. So it builds - there's a table in the NAT router which has outgoing traffic exits, egresses from the ISP. It translates the port number and puts it in a table so that, when the traffic comes back, it's able to determine which IP behind the NAT router should receive the traffic.

Well, that works well if you have a handful of machines. But remember the port numbers are only 16 bits. And not all of them are available for translation. Generally you use a subset of those. So if you've only got less than 16, you have fewer than 16 bits of port number, there's no way to map that into as many as 24 bits of possible IP addresses behind the NAT router. What that means is you simply need many more than one Internet-facing IP address. But still this is not an intractable problem. I don't know if ISPs are going to do this. But, I mean, NAT has kept us going as long as we have. Some ISPs do put their customers behind NAT already. And it's certainly possible that more could in the future.

The problem with doing this, second part of Lucas's question was, you know, are there downsides? The problem is that we use NAT, we home users use it because it gives us very good firewall, stateful firewall-like protection. Unsolicited incoming traffic has nowhere to go because it's the outgoing traffic that creates the mapping for the traffic to return to the proper computer. Which inherently gives us firewall-like capabilities.

But many customers deliberately create static mappings through their NAT router if they want to run some sort of server. If they deliberately want to make services available, their own home network services available on the Internet, then they're able to do so. In some cases they will use a feature of the NAT that does route unsolicited traffic over to a given IP address, over to a specific machine. In another case they'll use static port mapping. The problem is you lose that ability if you're behind an ISP NAT because the ISP controls the NAT router. You have no control over the NAT router. So you would be on a private IP address, and you'd have no way ever of reaching machines in your home network from out on the Internet because unsolicited incoming traffic would be dropped by your ISP.

So there really is a downside to this. I've long wondered whether sort of low-end consumer Internet users might ultimately lose the ability to serve on the Internet. That is, they would be clients of the Internet, but inherently not servers. That would be a sad day, but we're going to have an interesting year here, and we'll have to see how it develops.

TOM: That's another creative way for ISPs to charge you. In other words, if you're the kind of person who just wants to passively surf and send email, you'd sign up for the NAT account. But if you've got UPnP in your Xbox, or you want to do some remote computing, well, you've got sign up for the extra tier.

Steve: Right.

TOM: And NAT was supposed to protect us from them charging us for these sorts of things. Question #4, Steven in Baltimore has a question about home networking. He says: My question is about WPA passwords and how it salts the password with the SSID. Right now I'm using one of your super long crazy passwords, and it's no fun to type with onscreen keyboards like on the Wii or Nintendo DS. I know that pain, man. I'm with you, Steven. I'm wondering if I can use one of the passwords generated from your site as the SSID instead of the password, and a smaller password, say like 20 characters? Would that offer any protection with the smaller password?

Steve: Well, that is a brilliant idea.

TOM: That's clever, huh.

Steve: I thought that very clever.

TOM: And then you don't broadcast the SSID? Is that part of this?

Steve: Right, right. So we talked about the way the WPA functions and the way it converts your password that you provide into the key which is used by the crypto. What happens is these guys who did WPA knew their security very well, much more so than the people who did the original WEP WiFi encryption that was so badly broken. They take your password and the SSID and merge them together and then hash them using SHA-1 4,096 times. They do it 4,096 times, not because that makes it more secure, but because it makes it much more difficult to brute-force. That is, you take the raw password, mix it with the SSID, and then there's a computational burden to hash that 4,096 times, rather than just, for example, hashing it once. So anyone who was trying to do brute-force attack would be forced to do the same 4,096 repetitive hashes in order to get the key at the other end that they could then attempt to apply to your encrypted stream to see whether it decrypted. So that's how the system works.

What Steve is asking is, hey, I've got a fancy password and a simple SSID, and those are being mixed together and then hashed. Why don't I reverse it and use a simple password and a wacky SSID because they're being mixed together and then hashed. Wouldn't that give me the same? It's a cool and clever idea. The problem is that just turning off the SSID broadcast does not remove the SSID from the packet stream. Anyone sniffing your WiFi traffic will be able to capture your network's SSID. Not every packet contains it, but various management packets do. And so if they were doing a sniff, which they would be anyway in order to determine whether they were able to crack this, they would get the SSID. So the fact that you've made it really long and gnarly won't help you enhance the security, and you'd still, the only real strength you have is unfortunately from your password strength.

On the other hand, 20 really bizarre random characters, it doesn't have to be 64. 64, I would argue, especially if you're using one of my crazy passwords from GRC.com/passwords, which is just absolutely high-entropy gibberish, 20 is still a lot to go through brute-forcing. Most people are going to use a dictionary attack. They are going to try something. But the fact that the hash requires this 4,096 iterations in order to generate the key, that creates enough of a computational overhead that, you know, I'm not saying 20 characters is enough. It depends up on how determined you are to have security. Maybe make it 40. But still, that's an awful lot of security.

TOM: And those onscreen keyboards are really the problem. It's not your password, it's the fact that those onscreen keyboards are so darn hard to use in the end there because, if you want to stay secure, it's - you only have to put it in once. It's not like you're putting it every time you turn on the Wii or something like that.

Steve: Right.

TOM: Question #5, Joel Oliver in Pittston, Pennsylvania says: Hi, Steve. Wanted to chime in to the fact that TRIM is also supported in Linux, as long as the distribution is running kernel 2.6.33 or later. All that needs to be done is to add "discard" to the drives option in the etc/fstab file. Also FreeBSD 8.2 and up supports TRIM in the UFS file systems. Thanks for the great podcast.

Steve: Okay. So I misspoke, everybody. And, yes, we were discussing TRIM last week, and I said that the only operating system that supports it was Windows 7 [buzzer sound].

TOM: Never say never, huh? Never say only.

Steve: I know that Linux supports it. I was only, in my mindset, I was thinking about, unfortunately, I apologize, Linux people, I was thinking about Microsoft and Apple, PCs and Macs. And so of those operating systems, only Windows 7 supports TRIM. It absolutely is the case that the Linux and some of the BSD OSes support it, as well. So I wanted to - I chose Joel's note, but I received about 50 different pieces of email from people saying, "Steve, I'm using it on Linux." And I thought, okay. So I wanted to acknowledge everybody who wrote to me, to thank you and to say, yes, everyone is right. TRIM support does exist outside of the small, well, not small, but the Windows 7 universe.

TOM: I usually do this with Opera, so I feel your pain. And then I get the 50 things, people saying you forgot Opera does this. Question #6, Russell Spitler in San Mateo doubts the usefulness of software-based tokens. He says: In Listener Feedback #109 you discuss and encourage a software-based alternative to the common one-time use

hardware authentication tokens. This really rubbed me the wrong way. Please correct me if I am wrong, but given my understanding of the authentication tokens, a software solution invalidates most of the security they provide.

The one-time use tokens work on three inputs: the current time, a crypto algorithm, and a shared secret between the token and the authenticating server in the backend. While you can be reasonably assured that the shared secret found in a tamper-proof hardware token cannot be compromised without your knowledge, the same is far from true with a software-based solution. By embedding the shared secret in a software solution you are compromising the basic premise of the token. Having worked for a number of years in software security in particular, it is certainly a tractable problem to extract the secret from the software. Use of software-based one-time use password generators is far from the same level of security as a hardware token.

Steve: Okay. You did that in one breath, too. That was very good, Tom.

TOM: Thanks. I was assuming Russell had also written in one breath.

Steve: I think he did. He was upset with me. Okay. So he says, "Please correct me if I am wrong, but given my understanding of the authentication tokens, a software solution invalidates most of the security they provide." Okay, Russell, I think you're wrong. But I also think you have a point. First of all, what I was referring to was that I had discovered that the VeriSign VIP service was now offering a BlackBerry applet, and there's also one for iPhone, and I'm not sure about Android. There either is, or there will soon be. But that would be in addition to either the one-time use credit card, the one-time password credit card, or the often-discussed football, which is a time-based system.

So I have a web page that wants me to log in, say I'm using PayPal or eBay. And I have to use my BlackBerry in order to - which is running software, which knows what time it is and is generating a six-character token from that. So if I was running software that was running on the PC, in the PC, where it's inherently on the same platform in the same world where malware might be crawling around, where some scripting on the page that I've gone on might have some way of getting loose and, like, figuring out what my software key is, I mean, I can theoretically understand what Russell means.

But I've got an entirely separate device, physically separate, different operating system, different architecture. I don't see any way for software that I'm logging into a site with to somehow leap across space into my BlackBerry. Now, okay. I mean, playing devil's advocate, if my BlackBerry were infected, then I couldn't trust the integrity of this VIP system. That is, the BlackBerry would have to be infected. It would have to be communicating then with some site that knows I'm going to be logging in with my BlackBerry.

Anyway, I guess my point is that it seems a real stretch. Is a software-based solution which is using a telephone, a smart phone as a hardware platform, is that fundamentally less secure than a hardware token? Yes. I would agree it is. However, what we're accomplishing with using a time-based, always-changing token is dramatically greater security than if we don't have it. So my feeling is, it might be theoretically somewhat less secure, but vastly more secure than not using it at all. So, and it cost nothing. It was free. I added it as an app to my BlackBerry. So, I mean, to me it's a huge win, one that I'm happy to encourage our listeners to use because the cost is negligible, and the benefits far outweigh essentially not using it, or even having it compromised, which would be the same as not having it at all. So you don't lose anything, and all you have is a lot to gain.

TOM: All right, I'm going to move this along so we can get our last three questions in here before we have to wrap. Eric in San Jose says: Hi, Steve. Love the show. Long-time listener. So my broadband finally got upgraded at work. It's fast. But after speed tests I found that at home my D-Link DFL 200 firewall router could only support 15Mbps down and 3.5Mbps up while also running a site-to-site VPN. So I decided to set up an Astaro Security Gateway at home - free license - to test the ASG's performance. With a very old Dell box I got 95Mbps down with Comcast. Only 3.5Mbps up, so uploads are throttled by my ISP. Well, it took a little bit of thinking, and I decided to upgrade the NICs in the ASG to gigabit cards. I am now getting 392.90Mbps down on my fastest run. Still only 3.5Mbps up.

I thought other home users might be interested in upgrading to an ASG after hearing this. Please, however, don't be a bandwidth hog in my neighborhood.

Steve: So I thought this was a great post for a couple reasons. It is absolutely the case that these little plastic box, $49 SOHO consumer routers are built with the least expensive technology possible. They get the job done, but it is very easy to overload them. For whatever reason, many of the things that I've been doing in the last year, when I was working with the DNS Benchmark and the spoofability test, one of the things I had to deliberately do was throttle the spoofability test at GRC and the DNS Benchmark, specifically so as not to overload these little consumer routers. So I've bumped up against their very significant performance and packet processing limitations myself a lot.

So the first part of this is what Eric did was, he said, hey, I think I should be getting better performance than I'm seeing with my - in this case, this was a D-Link DFL 200 firewall router. So he switched to an old PC, a Dell PC on which he loaded the Astaro Security Gateway. And lo and behold, his download performance went from 50Mb to 95, meaning that that little D-Link router was the choke point for him. And when he switched to a PC, which is a much, I mean, vastly more capable computer than what goes in one of those little blue plastic boxes, he saw a huge jump in download performance.

Then, even though he was at 95Mb, okay, so 95Mb is very close to 100Mb, which was the limit of his networking cards. So he did the next thing, which was to switch from 100Mb cards to a gigabit card. And not surprisingly, it turned out that it was the Network Interface Card, that the 100Mb card was the thing that was blocking him at 95, which actually is pretty good performance to get on a 100Mb card. So he went to a gigabit card. Now he's at 392Mb.

So a couple of really good lessons, and one is to consider whether your little, cheap, cheesy, $40 blue router might be causing you to lose performance, which getting any random old machine and setting it up as a network gateway using Astaro Security Gateway, might give you much better performance. It's definitely something to think about.

TOM: Very clever. Very brilliant. And very frugal.

Steve: Yeah.

TOM: Very green, re-using. Question #8 comes from Nick Jackson in Austin, Texas, says: As I was listening to the latest podcast about browser fuzzing - that's #285 - your discussion of IE, Firefox, and Chrome's different approaches to implementing a do-not-track system reminded me of one of my favorite episodes - and one of the most unnerving - #264, side-channel privacy leakage. You endorsed Firefox's approach to do-not-track, namely adding an HTTP header, and that certainly makes sense as a forward-

looking measure to ensure that web advertisers are technologically capable of honoring these requests in the hopeful event that legislation makes this mandatory.

I am not particularly a fan of IE. However, from your description it sounds like Microsoft's Tracking Protection List (TPL) approach works essentially like a blacklist for stopping the browser from visiting, running code, or accepting cookies from certain sites. So my thought was, doesn't Microsoft's approach work better as a deterrent to side-channel privacy leakage? In other words, under Firefox's approach, recalcitrant web advertisers may nominally support "do not track" headers by not using tracking cookies, but nevertheless continue to profile and uniquely identify you through a variety of other unique facets of your machine.

IE's TPL blacklist, it sounds like, would simply prevent connections from being made to these profiling companies at all, stopping the side-channel privacy leakage problem at its source by never allowing you to share potentially uniquely identifying information to third-party servers. In fact, I would guess that the best combination would be both HTTP headers and a blacklist, which many people, including myself, get by using NoScript and AdBlock Plus. Do you think Microsoft is onto something unique here that Mozilla and Google aren't trying?

Steve: Well, I thought that was a great question, and I included it here because I wanted to - we sort of covered this point in Episode 265, when we talked about this. But I did want to draw our listeners' attention back to the fact that, as Nick says, Microsoft's TPL, the Tracking Protection List approach, from what little we've seen and read about it - this is going to be in IE9 - from what Microsoft has said is that any sites whose URLs match the pattern matching of a TPL list will be prevented from, exactly as Nick says, ever going. So it's exactly as he says, the notion of essentially creating a blacklist.

Many users now, for example, use a hosts file, which they'll get from the Internet and maintain it. The hosts file contains a huge list of typically advertiser domain names which they use to short-circuit DNS lookups, just for example turning it into 127.0.0.1, which is your own local IP, which prevents your browser from ever successfully going out to one of those advertisers. This is very much the same in that it similarly would protect - it would allow IE to manage which sites it was allowed to go to and which it wasn't. And essentially, I don't know that I would describe this really as side-channel leakage, but he is right that, depending upon the way the legislation is written, if the advertisers still got you, even though you had a header saying do not track me, then you're still making contact with a site that you may rather not have any contact with.

I'm hoping that this TPL approach does make it into IE. If this becomes a standard, then I would imagine that Firefox and Chrome and the others would pick up on this approach because it is more aggressive than what they're doing; but if it's sort of managed for us by websites that we visit, saying, well, you have to be able to visit our advertisers in order for us to provide you with valid content, please click this list in order to add an exception to the TPL, if you agree to that, then that gives users a lot of control. I can see it potentially gets confusing. We're going to have to see how it all washes out. But I did want to sort of reiterate that it's something hopeful from IE. And because it's so good, if it ends up taking off, we may see other browsers doing it, too.

TOM: And our final question comes from James Daldry in Raleigh, North Carolina. And we're back to IPv6. He says he uses a Linksys WRT54GL router - I used to have one of those routers - which is upgradeable to IPv6 by changing to Tomato firmware. My problem is that every article I read extols the virtues of hanging your system's bare IP stack out on the public Internet, which of course you could do since there will be plenty of IPs available under v6. But no one mentions the security benefits of NAT. So will my

router stop NATing under v6, and will I have to become a sysadmin for real? Or will the inside of my network remain the same, with 192.168 IPs? I'd hate to have to toss my Grace radio because it won't work with v6. I think this is a point of confusion for a lot of people with IPv6. Because it has so many addresses, it doesn't need nor does it use NAT. There's some other security considerations here.

Steve: Well, yes. And, first of all, everything is a confusion for everyone with IPv6. And he raises a really good point in his posting. He says he's got an appliance, this Grace radio, that doesn't support IPv6. Well, it probably never will. And there's all kinds of stuff, like Internet streaming radios and Roku boxes. Things that are state-of-the-art, you may be able to upgrade them and get newer firmware for them. But we've got, for example, I've got TiVos. All of my TiVos are IPv4, and they're using old hacked kernels. They're never, ever going to get upgraded. I know that. So I want to continue to have them on the Internet to get their directory information. And so what happens with v6?

Also, is it clear to us that ISPs are going to say, gee, how many IPv6 addresses would you like? The idea of not using NAT means that you would have to say to your ISP, gee, I'd like 25 IPv6 addresses. Well, is the ISP going to start charging you per IP? We've been there with IPv4. And the reason we're all hiding behind a NAT router is we want to look like we're just one computer. And the ISP knows no one is one computer anymore. But it works for them because they only have to give us one IP. So it conserves their IP space. That's not going to be a problem under IPv6. But then the question is, do we get blocks, do individual end users get blocks of IPv6 addresses? And if we do, what happens to our old IPv4 hardware? So, great questions. I don't have any answers yet. But it's going to be a really interesting next couple of years.

TOM: Can you make an IPv6 router that then does do NAT in IPv4 using tunneling?

Steve: Absolutely.

TOM: So there you go, I mean, hopefully somebody comes up with a few products like that. It sort of reminds me of the digital TV transition. You're going to have to get boxes for your old IPv4 stuff to convert them.

Steve: Exactly.

TOM: All right. Steve, it was great to be back on Security Now! with you. I really enjoy doing the show with you. Leo is out, everybody, on a cruise for the next two weeks. So I'll be sitting in on Security Now! for the next couple weeks.

Steve: Great.

TOM: Don't forget to check out GRC.com. You can find lots of excellent security, information, and products there. You find the transcripts there, as well; right?

Steve: GRC.com/sn, for Security Now!, will take you to a page with all of the past podcasts, where you can get the lower bandwidth version of Security Now!. I recompress it at 16KB and serve it myself for people who like the smaller, quarter-size file. The TWiT website and the podcast download, 64KB. And then we've got Elaine, who does a transcript in three different text formats for anybody who wants to read along.

TOM: That compressed version's going to be handy for Canadians now with the new CRTC ruling. They've got all those 25MB caps coming next month.

Steve: Oh, boy.

TOM: So, good to know. Thanks, Steve. Thanks, everybody, for watching. We'll see you next time on Security Now!.

Steve: Talk to you then, Tom. Bye.