Listener Feedback #163

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-395.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-395-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson's here. We'll talk about Microsoft's new Patch Tuesday. Wow. A USB exploit you just won't believe, and it's been in every copy of Windows until now. Plus, your questions and Steve's answers. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 395, recorded March 13th, 2013: Your questions, Steve's answers, #163.

It's time for Security Now!, the show that covers your security and privacy online with this cat right here, our Explainer in Chief, Mr. Steve T. Gibson. The "T" stands for Tiberius. He is...

Steve Gibson: That's going to really mess up my Wikipedia page now, Leo. People are going to scramble over there and make changes.

Leo: They're going to put it. It's not his middle name. Do not change his Wikipedia entry. Thank you very much. Hi, Steve. How are you?

Steve: Hey, Leo. Great to be with you again on Pope-Choosing Day.

Leo: We're waiting.

Steve: I guess the smoke is white.

Leo: A new Pope to emerge any moment.

Steve: Yup. And this is 31313. I noticed that as I was assembling things. I thought, oh, I like alliteration, we know. And so 31313.

Leo: And the day before Pi Day.

Steve: That's correct, 314. Well, yeah. I guess we really need two more years, 31415.

Leo: Oh, Pi Day 2015 will be a big one.

Steve: Oh, baby.

Leo: 31415, yeah. 3.41559. So at 9:00 a.m. in three years, two years, we're all going to be so excited for a moment.

Steve: We will have special coverage in case anything strange happens.

Leo: Special coverage for Pi Day. When is - the next one is 2038, isn't it, for Linux, for UNIX, the next big Y2K.

Steve: Yes, 2038 is when the 32-bit UNIX time wraps around to zero, and the world as we know it comes to an end.

Leo: Well.

Steve: Actually, it's already been extended to 64-bits.

Leo: It goes to 64 bits. Doesn't come to an end, it just goes to 64 bits. So what is our time - oh, we have a Q&A today, don't we.

Steve: It's a Q&A today. The week since we last spoke has provided us with ample entertainment and shenanigans and updates and things to talk about. And because of startup delays, we'll do as many questions as...

Leo: Oh, no. We've got time. Let's just - my lateness should not affect your content.

Steve: Yeah, but we don't have to push it all downstream. So you can decide, my friend.

Leo: So let's talk about Patch Tuesday yesterday, Second Tuesday of the Month.

Steve: Okay. So what was revealed, and this wasn't - there were three privately reported vulnerabilities that Microsoft closed. Among the 20 that were closed yesterday that were aggregated into seven sets of patches that pretty much covered the works - Windows, IE, Office, Silverlight, and SharePoint across all applicable operating systems - four of those sets were rated critical. And what's interesting is the most interesting one was only given the rating of "important," even though you gasp when you hear about it. But that's because, to get a critical rating, a vulnerability must be remotely exploitable with no user intervention. So it must be, as we've used the term before, "wormable," where, if a worm got loose on the Internet, it could reach into machines and propagate, as we have seen with Code Red and Nimda and so forth in the past. So those get "critical." The one that's interesting they only gave an "important" to because it's a physical presence hack, yet it is amazing. Now, in Microsoft's typical...

Leo: "Physical presence" means you can't do it unless you are actually at the workstation, at the computer.

Steve: Yes. Yes.

Leo: It's much less of a vulnerability.

Steve: But what it empowers people to do makes you gasp. And so in typical, you know, they copied and pasted something they wrote four years ago, and they keep repeating it. They say - the title of this was "Vulnerabilities in Kernel-Mode Drivers Could Allow Elevation of Privilege." Which is like, oh, well, that doesn't sound good. But, okay, elevation and privilege? Okay. So Microsoft says in their blog: "Today we are addressing a vulnerability in the way that the Windows USB drivers handle USB descriptors when enumerating devices."

Leo: Uh-oh.

Steve: Okay. So that means - so the drivers are in the kernel. USB descriptors is not part of, like, the file system on a USB drive. Those problems we've seen before. The USB descriptors is down in the USB protocol stack, and it turns out there were bugs there.

So continuing, Microsoft says: "This update represents an expansion of our risk assessment methodology to" - what BS. Oh. Anyway, "to recognize vulnerabilities that may require physical access, but do not require a valid logon session." Okay, now, there was no such thing before, but this created that methodology for this reason. And they say: "Windows typically discovers USB devices when they are inserted or when they change power sources, e.g., if they switch from plugged-in power to being powered off the USB connection itself. To exploit the vulnerability addressed by MS13-027, an attacker could add a maliciously formatted USB device to the system." Which means plug it in. "When the Windows USB device drivers enumerate" - which means, you know, scan...

Leo: Say what have we got here, what have got here, yeah.

Steve: ...the device, yeah, what just arrived? And you always hear that ga-glook sound, when Windows says, oh, someone just plugged something in.

Leo: [Vocalizing]

Steve: Yeah, exactly. So when they "enumerate the device, parsing a specially crafted descriptor..."

Leo: Oh, lord.

Steve: "...the attacker could cause the system to" - and again, could, yes, uh-huh, can - "cause the system to execute malicious code in the context of the Windows kernel."

Leo: Could, but only if he wants to.

Steve: Yeah. Only if that was his intention.

Leo: Only if he chooses to.

Steve: Only if he designed his USB device to do so. "Because the vulnerability is triggered during device enumeration, no user intervention is required. In fact, the vulnerability can be triggered when the workstation is locked or when no user is logged in, making this an un-authenticated elevation of privilege to kernel level for an attacker with casual physical access to the machine. Other software that enables low-level passthrough of USB device enumeration may open additional avenues of exploitation that do not require direct physical access to the system."

So what all this means is that there has been a problem, which this patch fixes, such that it was possible to exploit a defect in all Windows operating systems across the board, for all time, apparently, where you simply plug something into a machine, whether it's in use or not, just if it's powered up. You don't have to be logged in. It could be a server where no one is typically logged in at the console. It could be a machine which has been locked, and so you've got to, oh, type in a password in order to access it. No, not necessary. You just slip this little thumb drive in the side of the machine, and you take it over. So...

Leo: That's amazing. That is, I mean, that's...

Steve: Not critical, Leo. That was not critical. That was just important.

Leo: But of course what it means is somebody could create a thumb, could have a thumb drive and go to a library or somewhere with public computers, and boom boom boom boom boom, just take them over.

Steve: And what it means also is that what we're seeing now is all these patches are being immediately reverse-engineered. So while this was privately reported to Microsoft, and there wasn't apparent, there wasn't known exploitation of it in the wild, that will exist tomorrow.

Leo: Does it need autorun? I mean, no, because it automatically runs these drivers when you install, when you stick it in.

Steve: Yes. This is in the - at the kernel level, not at the file system autorun, "what is that" kind of parsing. We keep - we've had all of those. And so finally Microsoft gave up and just turned that off under duress because, oh, that was going to make it harder to use their operating system. So, no, no, this is just - at the low level, at the hardware level, the Windows hardware drivers see that something has appeared, and in querying it for its specs it has a way of misusing the driver, essentially a buffer overflow-style attack in the kernel that allows it to just take over the machine. And of course it's got the rest of the drive right there to provide all the files and malware that it needs to squirt in while it's doing it. So, yeah. So this is one you want to update. Not that there are any that you don't, but it would be good to do this and not wait long because there will be - this is - people are going to be having fun with this. Meanwhile...

Leo: Well, there you go. There you have it.

Steve: Not to be left out, Leo, the fourth update in as many weeks of Adobe's Flash. We've had basically one a week.

Leo: Flash is in a race with Java to see who can be compromised most in the month of...

Steve: Yeah, actually I think Java may be inching ahead. But we'll talk about that next.

Leo: Flash is catching up, though.

Steve: Yeah, so we're now at 11.6.602.180 for Windows and Mac platforms, as I said, the fourth update in as many weeks. This addresses four known security flaws in Flash Player. But unlike the previous emergency Flash updates, Adobe didn't wait, which is probably wise because, as we know, last time they waited, and then they thought, oh, well, we knew that this was a problem, but we were going to be updating when we could. So this time they're just pushing it out immediately. IE10 and Chrome both update automatically. Firefox is getting better about that. Safari has pretty much locked it down so you're less exposed, with that and with Java. So we're seeing the industry adapting as well as it can. And unlike Java, which fewer people have a need for, anyone who uses iOS

devices is running into the annoyance of not having Flash on iOS. But the flipside is you don't have the security problems.

Leo: Even Adobe has stopped making a mobile Flash even for Android. So it really - Adobe says you don't want it on a mobile device. But you do kind of sort of have to have it on a desktop computer still.

Steve: Yeah. I do think I like what - I'm still at - Firefox is my primary browser, and with NoScript. And it has this stuff locked down. So that unless I am somewhere where I trust, and I really know that I want to, like, play Flash, then I click, it's the click-to-play approach where I click on it, and then it enables Flash...

Leo: That's probably the right way to do it, yeah.

Steve: ...on that page. I really think that is. That way you're just - you're less prone to - the normal approach is Flash is embedded non-visually on a site. That is, some links are changed, and there's just a little Flash plugin embed that isn't like, here, play. Click here to play this video. It just sneaks something by on a site that doesn't otherwise - isn't obviously a Flash host. And so anything that requires click-to-play Flash completely protects you from that. So, and NoScript and Firefox, of course, are among those.

But as I said, Java is not to be left behind. Many people tweeted me the link to the new "days since the last Java vulnerability" site. And I can't remember what - it's like javazeroday.com or something. But the point was I got a nice tweet from a Michael Horowitz in New York who said, "Steve, just an FYI. There are now, as of March 8," so that was Friday, "12 known and currently unpatched bugs in the Java software released all of four days ago." Which we talked about last week, that that was Version 7 Update 17. Like seven new ones were found since then. Our friend Adam Gowdiak has been busy.

Anyway, Michael has a site that looks kind of nice. If you're somebody who's stuck using Java, you have to use it, you might want to check out JavaTester.org. That's Michael's site. I went there. It looks nice. He does some testing of your browser, seeing how, like if Java is truly disabled in it. And he's done some digging into how the various browsers handle that setting in the Java setup where you can say remove it from my browser. I updated, and I went through, in Windows, through the Windows control panel, to the Java applet.

And on the first page that comes up of the tabbed dialogue, it says "Java is enabled in this browser." And it's like, oh. And then it's like, "Go to the Security tab to change this." So you go to the Security tab, uncheck that "Enable Java in the browser," and then I went back to the first page, and it said "Java is not enabled in the browser." But there's been some concern that it seems to be reenabling itself because I had turned it off before, and it turned itself back on with an update. So it's worth doing after you update until maybe they will start disabling it by default and only turning it on when people need it.

But so these are now known bugs in the latest Java. And we imagine that they will be surfacing in the wild before long because that never seems to take very long. And I have a note, somebody suggested another acronym for Java that was kind of clever. But so far Just Another Vulnerability Announcement seems apropos...

Leo: That's pretty good.

Steve: ...to this one, yeah. And several people tweeted that the issue of Apache and the Do Not Track politics with IE10 had been solved, or resolved. And so we've talked about this many times. And the source code that was in the config, the HTTPD.conf file was commented as "Deal with user agents that deliberately violate open standards." And my upset over this was that...

Leo: In other words, Internet Explorer.

Steve: IE10, yes.

Leo: I.e., IE.

Steve: Yes. Or e.g., IE. So on the GitHub site there is some dialogue about this. And the anchor posting I liked because it was one of the main thinkers about this who said this is exclusively regarded - oh, I'm sorry, that was on the Apache.org Bugzilla site - said this is exclusively regarded as a bad idea because "Apache is the wrong point to add this setting. DNT does only matter for a very small subset of servers, i.e., those of ad agencies. So this will, one, result in unnecessary processing for most installations; and, two, the DNT header will only be processed at a higher level, e.g. PHP, where this check could be made. With this setting, Apache is 'stealing' information from the actual processing logic that will handle the DNT header." Which is a good point because Apache is not removing it unless it sees that your user agent is IE10. So, and this was my argument, was that it shouldn't be at the server. It ought to be at the application level.

So then he says, second point: "This is damaging to the reputation of the Apache project as it shows people hiding their personal political agenda in open source software." And he wrote, "German publisher Heise is already reporting about this." And he said, "Using this setting would mean also people who explicitly decided to use DNT will be ignored, which is a much worse case than using the DNT setting from people who don't care." And he said, "Whether DNT should default to 0 or 1," and he said, "Let's face it, there will always be a default unless you require the user to set this on first launch of the browser with only the buttons 'Enable,' 'Disable,' or 'Quit Browser,' which will never happen," he said, "is something to decide by the specification, not by Apache." And finally, "Unlike what the comment says, IE10 actually asks the user about DNT when Windows 8 is installed, as shown in the link thread above, making the point of this commit invalid."

Leo: It does ask, but it asks that with a number of other things in somewhat fine print. So...

Steve: True, yes. Anyway, so a number of people got onto the thread, had a very nice discussion. And essentially Roy, who committed this to the config file update, lost, and it is now - it's still there, but all commented out.

Leo: Interesting, interesting.

Steve: So it is gone by default from Apache. But if your own feelings said I want to enforce this against an IE10, well, it's there, and you can take the comments out and the work is done for you.

Leo: And this, by the way, is why I like open source software, because this discussion happened in public, and everybody had a chance to speak. I think they came to the right conclusion.

Steve: Well, and in fact our friend at Stanford, whose name is escaping me, I referred to him last week, he's the privacy and tracking guy who's part of the W3C anti-tracking committee. His responses in this thread were really well informed and well thought out. And he was able to bring the exact perspective of the committee, saying that this is not what we want to do. It is not a non-compliant browser. Those decisions have not yet been made. So it's like this was, if anything, I mean, the least it was was premature because this was jumping the gun relative to where the current thought is on the whole issue. So again, to me, this is all interesting stuff because it's evolving, and it's what's happening right now. How do we deal with the tension that exists between privacy on the Internet and monetization on the Internet? And those are two forces that are in tension with each other.

Leo: So my question, actually a good question from the chatroom is, is there any way to know if a browser - if the Apache has been compiled with this on? I mean, I guess not. I mean, you could look at the traffic, I guess. I don't know.

Steve: Good question. I don't remember what the patch does. I have it here. Let me scroll back to it. Does it fake the...

Leo: I don't know. Or ignores it or...

Steve: Well, it apparently, if you have IE10, it changes the header to say that it is unset. Meaning that there has been no decision made.

Leo: Regardless of whether it's been set.

Steve: Correct. So whether...

Leo: I guess you could if you looked.

Steve: That was one of the other points made further down in the thread is that, if a user did deliberately set it or unset it and was using IE10, then Apache was, if this was in place, deliberately ignoring their explicit choice.

Leo: Right.

Steve: So Apache has no way of knowing whether an IE10 user did or did not want tracking. Even if they did or did not, Apache was not going to provide that information up at the application level running on the server. So just all around a bad idea. That's not the job of the server. That was my technical, from a technical standpoint, independent of how you feel about it, it's not the job of the server. The server's job is to make the connection, provide the information to the app running on top of it.

Leo: I have to say I agree.

Steve: Yeah. Bruce Schneier's famous comment, "Attacks only get better, they never get weaker," got a little workout also. A group of cryptographers, led by - I don't know if it's led by Dan Bernstein, who's a world-famous cryptographer we've spoken of often. He's got that neat site, cr.yp.to. I just love that.

Leo: Say that again?

Steve: Cr.yp.to.

Leo: Cryp dot to.

Steve: Crypto. No, it's crypto with dots every other letter. So, very cool.

Leo: "To" is Tonga. So you have to go to Tonga to get that.

Steve: And then convince them that you want a two-letter domain, "yp."

Leo: Yp.to. And then you create your server to have a "cr."

Steve: "Cr," exactly.

Leo: Love it. That's Bruce.

Steve: Okay. So there's a lot of history here, and this is interesting. We've known for a long time that there's a weakness in the RC4 cipher. To remind our listeners, RC4 was what WEP, the early original WiFi cryptography or WiFi crypto used as its cipher. RC4 is itself a very good cipher. It uses two arrays of bytes, two 256-byte arrays of bytes. And when you give it a key, the key schedule process or key setup, as it's called, initializes - wait, is it two arrays? No, it's one array and two pointers, I'm sorry, because I've written some RC4 myself. So it's one array, very lean, one 256-byte array of obviously 256

bytes.

When you give it a key, it scrambles those in a defined fashion from the key. So that's its known starting condition. Normally you just fill the array from zero to 255, and then you apply the key, which - and what it does is it employs a series of swaps. So the two pointers are the places where you swap the bytes with each other. And so, given a key, it initializes this in a scrambled, but deterministic way, so the same key always starts with the same scramble. Then, when you're using it, what it is, it generates pseudorandom streams. So it's a - RC4 is a stream cipher. It emits a pseudorandom stream of bytes.

And we know that, if you take plaintext and XOR it with pseudorandom data, you get gibberish. You get pseudorandom data. You get something that, surprisingly, is really good encryption. And then, on the receiving end, you apply the same key to the same algorithm. You get the same pseudorandom stream. You XOR that against the already XORed data, and out pops the original plaintext. Very cool. The problem with RC4, though, is that, due to the simplicity, essentially, of this approach, there were first of all weak keys, some keys that you could give it, and you can almost sort of intuitively get that. Some keys that you would give it would not do a very good scrambling job to initialize that array. And so that was a problem. There was the weak keys problem.

And the way the WiFi system worked back then, WEP, is that there was an incrementing value that incremented over time, and it could wrap around. So if you were online long enough, and it wrapped around, you would start reusing the same pseudorandom data. You'd generate the same pseudorandom data, and that's a big no-no. You never want to do that with a simple XORing cipher, or it's an immediate weakness.

The other problem was that it needed - RC4 needed more time to warm up. That is, as you used it, it continually rescrambled. So the key initially scrambled that vector of bytes. But then, as you used it, it kept scrambling it even more. Well, it turns out that the initial setup, the key setup really - there were some weak keys, but also there was actually some predictability, no matter what key you used, in the initial key stream, as it's called, the initial stream of pseudorandom data. After about 256 bytes had been emitted, then even that was lost. That is, it really became robust.

So, and I've mentioned this before, that all we really needed to do to fix RC4 was warm it up. Discard the first 256 bytes of data. So change the way you use the cipher so that you use the key, then you run it for enough to get 256 bytes of data out, that would take no time at all. Then you've got a much better key stream. And that's one of the ways that WPA fixed this problem. WPA can use AES, but also RC4 because there's nothing, if you use it right, there's nothing fundamentally wrong with it, although people could also argue that putting all those requirements on the way it's used really represents a problem for it. And it's like, wait, you ought to just be able to give something a key and start using it, rather than, oh, well, you've got to avoid bad keys, and you've got to warm it up for a while.

So what happened is these crypto guys just - and this will be in a paper coming out shortly - completely characterized the nature of the initial weakness in RC4. That is, it was known that some of the bytes weren't as random, no matter what the key was, as they should be. And that made people uncomfortable, cryptographer people. But they said finally, okay, let's figure out exactly what this means. They nailed it. So that the beginning of RC4 is now very well understood.

And this is a perfect reason why Bruce Schneier's famous quote, "Attacks only get better, they never get weaker," applies. It's because over time we develop more knowledge of the way these - what seemed like it's, oh, really mysteriously complex initially, like we

keep knocking off hashing algorithms one after the other because people really smart who have the focus are able to just chip away at it, a little tiny bit, just by digging deeper into it. Suddenly the bits don't all seem so random. It's like, wait a minute. There's a little pattern over here. What does that mean? And they go off, and then they come back with, uh-oh, this means something.

So here's the point. Because of the attacks on SSL, the most recent and popular attack is the so-called "BEAST" attack, B-E-A-S-T. And this was done by those two guys we talked about. One was on the beach somewhere looking at bikinis, I guess, and the other one was home. And they were corresponding, and they realized that, because of the use of initialization vectors in the cipher block chaining that AES was using, or any other cipher that uses cipher block chaining in SSL, there was a way that you could mess with encrypted traffic and over time gain information because the chain went from the end of one packet to the beginning of the next. That is, the chains were linked across packets. And that allowed somebody who was able to inject and intercept traffic to fuss with that linkage and play with the cipher in a way that caused some information disclosure.

So the consequence of that was that the recommendation has been don't use CBC-based ciphers preferentially. Use RC4 because RC4 works completely differently. It's not a block cipher. It's the block ciphers that need to use cipher block chaining as their algorithm in order to do crypto so that the blocks are interdependent. We've talked about this a lot. So, for example, when I was setting up my new servers just a couple weeks ago, there was a site, SSLLabs.com. I recommend it to everybody. Go check it out. It's very interesting. SSLLabs.com is a site which will check the SSL security of any server you ask it to. It goes and performs a series of SSL or TLS connections to the server.

And we've talked about how SSL works. Normally the client who wants to connect gives it a whole list of ciphers, and then the server picks one and then says in its return handshake, here's the one I want to use. So it's easy, for example, to determine which ciphers a server supports by only giving it one. Just give it one. And if it comes back and says I can't do that, then you know it doesn't support that cipher. And then you get a different one, and it says, oh, yeah, okay, I'll do that, if you're only giving me a choice of one.

So the point is it's possible to remotely probe a server and learn everything about the crypto that it supports. SSLLabs.com does that. Because I was using Windows 2000 from the Dark Ages, I used to get a "D" there, and users used to, I mean, I've had people, podcast listeners who'd say, hey, Steve, have you gone over to SSL Labs and looked at what GRC.com gets? And it's like, yeah, yeah, yeah, I know. And it was because the crypto stack in Windows 2000 was from Windows 2000.

Leo: It was just old.

Steve: It was just old.

Leo: Not broken, just old.

Steve: Not broken, and no one was ever really in any danger. But, again, these are theoretical attacks; and, as we know, they only get better over time. So one of the reasons that I was really glad to move to Server 2008 is I got a state-of-the-art updated stack. And in fact I went to 2008, and it still did not support TLS 1.1 and 1.2. It did

support TLS 1.0. So then I jumped to R2, and because I thought, if I'm doing all this work, I don't want to already be behind on day one. So for once I actually got current. And I went to R2.

Leo: There is an argument for being current, isn't it.

Steve: Yeah, there is. And now GRC gets a Grade A.

Leo: Yay.

Steve: And but not 100 percent on a couple. And I thought, wait, why am I not getting 100 percent? It turns out that this guy is so picky that you've got to have 4096-bit or larger keys. It's like, okay. So nobody has that. But he is being future-proof, so I understand that. At least I get an "A." But I didn't at first. When I first set up my servers, he was...

Leo: Because of TLS.

Steve: Well, because of BEAST. I was vulnerable to the BEAST attack. And I thought, okay. Well, I knew what that meant, of course. It's because the default ordering of the ciphers was such that RC4 was way down the list. And the idea is that the server chooses in the order of what it considers greatest security, but you've got to tell the server what the better ciphers are. So the default installation of Windows gives you a bad grade on SSL Labs because RC4 has pushed way down.

So because I wanted an "A" at SSL Labs so nobody would worry about GRC.com, I did what everyone does, and the only thing you can do is move RC4 to the top. You have to have it first. All the advice you can find on the 'Net is, if you want to avoid the BEAST attack, and it's not just SSL Labs, I mean, it's actually to avoid the BEAST attack, the server has to say "I want to use RC4." As a consequence today, 50 percent of the encrypted traffic on the Internet is using RC4 because of the BEAST attack. Even though there are many, many better ciphers available, we can't use them because the later ciphers, TLS 1.1 and 1.2, they fix this problem, except that clients haven't been updated. So servers, like my server happily has the very latest cipher suites available. But if the client can't use them, then it knows about RC4. And that's better, some would argue, than using one of the cipher suites which can be compromised by BEAST.

So what all this means is that the paper that has come out that has fully analyzed the first 256 bytes has identified the way that those pseudorandom bytes are much less random and much more pseudo than we wish. The consequence of that is, if you were to passively monitor encrypted traffic, where the same traffic was being repeated over and over and over, you would see the beginning of that connection as it's coming up. And if you were to capture the first 256 bytes of that, and we're talking, I think it was four to the - I remember seeing, no, I'm sorry, two. I remember seeing a 2^32 and a 2^20-something. Anyway, it's millions. You have to get millions of samples. But if you do, you can figure out the plaintext. You can decrypt that first blob of traffic. And the reason that's a concern is that browsers generally have query headers at the beginning of their query. And the query headers contain the cookie which is your authentication. So again - we're having to follow a long chain of what-ifs, I realize. But again, these things only get

better. Somebody will have something that runs in Starbucks before we know it.

Leo: Don't feel bad, by the way. Bartman in our chatroom says IBM.com gets an "F." So you're in good company. They're probably running an old server, too.

Steve: So, and, see, now I'm kind of wondering, well, what should I do because I don't want this. So, I mean, there are proposals for fixing this. The problem is everybody would have to update their clients. All you would have to do, for example, is discard that first 256 bytes out of the RC4 stream cipher, as I mentioned before, and then the bytes are sufficiently pseudorandom. And it's interesting, too, because there's a chart - I meant to tweet these links, but I didn't. But if you - probably if you Googled "Attack of week: RC4 is kind of broken," I think probably if you Googled "Attack of week: RC4 is kind of broken," that's a great blog posting that links to the page where there's a chart showing the 256 bytes, each of the 256 bytes across the horizontal and the probability...

Leo: It's from CryptographyEngineering.com.

[blog.cryptographyengineering.com/2013/03/attack-of-week-rc4-is-kind-of-broken-in.html]

Steve: Yes, that's where it is. And then it shows you, like, a graph that spikes in sort of like a comb fashion, where the comb teeth are shortening as it goes further out. So the bytes earlier are more easily determined than those later. But what this ultimately allows you to do is theoretically get somebody's authentication cookie from their browser headers, if you could induce millions of requests. And their point was that there have been things that do that. For example, the BEAST attack works by injecting some JavaScript that causes your browser to make the kinds of queries that the BEAST attacker wants made. And you could also inject some JavaScript that would cause your browser to make millions of repetitive queries to Google.com, for example, which each one of those would carry your currently logged-on Google.com cookie, if you were logged onto Google.

So anyway, an interesting piece of new, state-of-the-art cryptography, and sort of a problem here because the industry, in trying to thwart the BEAST attack for currently supported ciphers, has elevated RC4 as preferential to any of the block ciphers, which really are better ciphers, because of the way SSL 3.0 and TLS 1.0, the way they chained their packets together with cipher block chaining. This has been fixed in TLS 1.1 and 1.2, but not everybody's using it yet. So, interesting dilemma, and cool stuff, just from a theoretical standpoint.

Speaking of cool stuff and theory, a little glitch hit Bitcoin just yesterday. And what happened is really interesting because all Bitcoin implementations up until 0.8 used the Oracle Berkeley DB backend as its database. So the bitcoin mining and the Bitcoin clients and everything, they were using this Oracle Berkeley DB 0.7. The developers decided to switch, with the release of 0.8, to Google's own database, which is called LevelDB, very nice, slick, clean. It's a key...

Leo: Yeah, so no SQL, so it scales beautifully across massive installations.

Steve: Yes. It's very cool. It's a key and value database, so you're able to store basically keyed values. You say here's a key, here's a value; here's a key, here's a value. And then later you say what's the value for this key, and it's able to find it, just bang. What happened is there was a never-known bug in the old Berkeley DB, and everybody was using it with the bug. The bug kind of kept the playing field level. But when people began switching to 0.8, which should have been absolutely compatible, different things happened, and Bitcoin broke.

Essentially - and we did a complete podcast on Bitcoin [SN-287]. Anybody who wants the background, I'm not going to go over the whole thing again now. But it's really neat the way it works. I mean, the technology is so cool. But essentially it caused a fork in what normally is a continuous movement of work of bitcoin mining. It diverged the bitcoins and caused a huge upheaval yesterday. I mean, they realized something was broken. I read up on all of the current discourse that I could find. And, I mean, the gurus are still saying, okay, we have to understand exactly what happened. And what they're telling people is that miners should immediately go back to 0.7.

They're basically going to have to kill off some of the forked work that was done and then figure out what to do about, like, moving forward. Do they, like, they need to exactly character- they don't even, as far as I know it, don't even yet know exactly what the problem was. It involves locking. There are things called "locks" in a database which you do in order to prevent data from being modified while you're in the process of modifying it, so that there aren't collisions. It's the old case where two people each read something, and then they, like, increment it or modify it, then they both write it back. Well, the first person to write it back gets what they wrote overwritten by the second person to write it back.

So instead, if you know that you're going to modify something, you lock it so that it's yours exclusively. You then read it, modify it, and write it back. Anybody else who's trying to get to it at the same time, they have to wait until you release your lock, and then they're able to access it. So this is well understood technology for multithread, multiprocess, multiuser databases. And there was a limit on locking which 0.7 had, and everybody had the same limit imposed because everybody was using the same database. LevelDB is better and doesn't have this imposition. So its behavior subtly changed, and that broke Bitcoin.

Leo: Boy, it just shows how complicated this stuff is.

Steve: Oh, my goodness, it's complicated. Yeah, I mean, and you're reading through this dialogue of these gurus, and your head just spins. It's like, holy criminy. It's, like, just amazing.

Leo: And I can see why they can't just patch it right away, because they have to understand what the heck's going on, make sure - it's complicated.

Steve: Yes. And work was done by...

Leo: Right. You can't throw out the existing work, or I guess they...

Steve: Yes.

Leo: ...might have to, but...

Steve: Yeah. Now, I did also notice that coins are - it used to be that when you successfully found the magic value to make the hash have all leading zeroes, which is part of the coolness of bitcoin, really any listeners who are more recent additions to our podcast, let me urge you to go listen to the Bitcoin podcast [SN-287]. Back then I did a pretty good job of sharing my love of Satoshi, who was the guy who developed this. And it's just immaculately conceived. It's amazingly cool.

Leo: Not an endorsement, I should point out, not an endorsement of the idea of Bitcoin, but merely of the technology of it. We don't...

Steve: No, but what I will endorse is the fact that this thing has staying power, Leo.

Leo: Yeah, yeah, that's true.

Steve: I think it's here to stay. I think the world, the governments may not like it.

Leo: An extra-governmental currency.

Steve: Yes, a virtual Internet currency. I think there's a place for it. And I think Bitcoin is going to be it. So anyway, just it is incredibly cool technology. And I fired up Bitcoin and played with it for a couple weeks, and I minted a coin back then. I got 50 bitcoin, which apparently is worth about $2,500 now.

Leo: Wow.

Steve: So, yeah.

Leo: But, yeah, to try and spend it anywhere that...

Steve: Oh, no. You're able to use any of the exchanges and turn it into cash.

Leo: Actual cold hard American dollar?

Steve: Yup.

Leo: Wow.

Steve: Yup. So back then a single bitcoin was - a single hashing success minted you 50 bitcoin.

Leo: Not anymore. Not anymore.

Steve: But that drops in half every four years.

Leo: Right.

Steve: So it has been that four-year gap since then. Bitcoins now, when you make a successful hash, you get 25 bitcoin. And...

Leo: We should point out you were fairly lucky to get that. I mean, the odds against it are high.

Steve: Well, and, boy, since then - this was years ago. And this was just on - it was on this little machine, a little i7 quad core. And it just kind of - I looked at it one morning, it was like, oh, look, I got one. Well, now, I mean, people have living rooms full of Freon-cooled, ASIC-driven, insane monster caching bitcoin mining machines. And the Bitcoin system automatically scales up the difficulty in order to keep the coin creation rate set per its algorithm. And that's asymptotically approaching a final maximum. So since then, you're not in the game if you just run a computer and say, oh, maybe I'll get lucky. It's like, no, no, no, honey, that day is way past. But very cool technology.

I missed a note from our Mr. Wizard, who did another Security Now! video. That is, I missed it because he sent me email on Wednesday when I was in the middle of prepping for last week's podcast. Now, since then, there's been another one. So I just want to remind people about AskMrWizard.com. You can go to AskMrWizard.com, and you can click on the Security Now! link, or you can do AskMrWizard.com/securitynow. He's got from Episode 3 "NAT Routers as Firewalls." And these, I should say, these are animated videos to complement our podcast. And they have our audio, but he's added video to it. No. 8, "DoS Attacks." He just did the "Rootkits," No. 9 on rootkits. No. 10 on "Open WiFi Access Points." No. 42, that is, Podcast 42, on "NAT Traversal." And he also did one on a recent podcast, 388, on "Memory Hard Problems." So those are there.

Leo: It's pretty neat, yeah.

Steve: And I just had to also tell people about a piece of freeware that I've now been using for a month that I really like on Windows called CCleaner.

Leo: Oh, yeah. Oh, yeah.

Steve: It's by a company...

Leo: Our audience knows about that very well.

Steve: Piriform.

Leo: Piriform, yeah.

Steve: Yup. I really like it. I didn't know it was known. I ran it on my own personal machine for the first time yesterday, or two - yeah, I think it was yesterday. It found 775MB of accumulated gunk. In fact, Google Chrome alone was responsible for 430 of those megabytes. So it's very cool. Also it found 1,126 debris keys in my registry. And when I removed all those - you have to do that successively because sometimes keys refer to keys. So the second pass it found 33 more, the third pass 11 more, and the second pass two more, and finally no more. One of the things I've noticed is that any old machine that you've had for a while, when you uninstall software, the uninstall system tends not to remove registry keys. So they tend to accumulate.

Anyway, this thing also allows you to view and play with your system startup and your browser add-on startup, your scheduled tasks, and it even has a free space wiper for people who want to wipe the free space on their drive. So anyway, it's called CCleaner. It's freeware. They're not trying to install any junk on your machine. It's not ad-supported or anything at all. Very, very clean, good stuff. So I wanted to just say, hey, I independently discovered it. And I'm glad you guys know about it, too, you and your team.

Leo: Oh, it's very, yeah, very popular amongst our crowd. I don't generally recommend it. Most of that data was just temp files, like the cache files on Chrome. And so that's fine. I mean, deleting temp files is harmless.

Steve: Yeah, I just like to...

Leo: The registry keys is a little tricksy. I found CCleaner is pretty conservative, so you're probably all right. But you've got to be careful because you can screw your system up.

Steve: And it does also, in cookie deleting, it apparently applies some sort of a smart cookie algorithm. So it will try not to delete the cookies representing your logged-on authentication, which is nice, instead of just doing a wholesale cookie expunge.

Leo: Right. By the way, I don't know if you saw the article in Ars Technica, but I thought this might be a good subject for a future show: "Meet the men who spy on women through their webcams." We've talked about the idea, the theoretical idea that you could access people's computer with a rootkit or something. And apparently there are fairly easy tools, RAT tools they call them, Remote Administration Tools.

You remember Back Orifice, back in the day? Well, they've gotten much more sophisticated and apparently quite widespread use. So I have, on the radio show, we periodically get calls from people saying I'm sure there's somebody in my computer. Well, I guess it's more widespread than we thought. There are forums where people share pictures they've gotten from people's machines, and of course turning on webcams and things like that. So I don't know. I'll send you the link. It's at Ars Technica in their Tech Policy Law & Disorder blog. RAT tools. And I thought this would be a good...

[arstechnica.com/tech-policy/2013/03/rat-breeders-meet-the-men-who-spy-on-women-through-their-webcams]

Steve: Yeah, remote access.

Leo: Yeah, this might be a good subject for you to probe.

Steve: And of course we've talked about the problem with cams, and I always say just stick a little - stick the sticky part of a post-it note over the webcam, if it's something that you're not using all the time, or just stick it over there when you're not using it.

Leo: It's funny that you mention that because here's the relevant paragraph from this Ars Technica article: "Welcome to the weird world of the ratters. They operate quite openly online, sharing the best techniques for picking up new female slaves" - that's what they call them - "and avoiding that most unwanted of creatures, 'old perverted men,' in public forums. Even when their activities trip a victim's webcam light, and the unsettled victim reaches forward to put a piece of tape over the webcam, the basic attitude is humorous. Ha! You got us! On to the next slave!"

Steve: Wow.

Leo: Yeah, pretty creepy.

Steve: I guess you could use frosted Scotch tape, come to think of it, although I like the reusable stickiness of a post-it note.

Leo: Yeah. You don't put it on the camera lens, just over it. Above it. Oh, lord above. And actually there are starting to be computers with little doors over the camera, which is just very cool.

Steve: Yes, that's so nice to have a shutter. You really want a little shutter.

Leo: Yeah, yeah.

Steve: And keep it closed unless you're actively using it. We got another tweet suggesting a Java acronym, for J-A-V-A, of course: Just Another Valueless Add-on. So that's pretty good, too.

Leo: Yes, yes.

Steve: And I don't have time, lord knows, to mess with this. But there is something cool happening that I just wanted to give our listeners a heads-up for those who would be interested. And that is, over on Kickstarter, there's something called the Mojo Digital Design for the Hobbyist. It is a hobbyist-grade, very nicely put together FPGA platform, Field Programmable Gate Array. A field-programmable gate array, I think it's using a late-model Spartan FPGA chip. I think $55 gets you, like, the whole kit-and-caboodle. They've already way passed the amount of money that they were hoping to raise, and they've made so much extra money, they've had so much more pledged, that they're adding new features and things now because they can afford to. But so there will be software for programming it.

A field programmable gate array is basically software-controlled digital design so you can design gates and logic and counters and a little simple computer, for example, all in software, download it into this thing, and it dynamically configures this chip. It has a ton of I/O, 50-some I/O, lights and buttons and analog inputs, and it just looks very cool. So for anyone who thinks that would be fun to play with, I wanted to give you a heads-up. Mojo: Digital Design for the Hobbyist is the link over at Kickstarter.

[kickstarter.com/projects/1106670630/mojo-digital-design-for-the-hobbyist]

And I found a really neat story, something I hadn't ever seen before, about SpinRite, where he said "Backups are not always practical," from Howard Matthews, obviously a listener. He said, "Years ago I landed a great job working for a well-known PC manufacturer" - he's not telling us whom - "in their R&D department. My job" - and this is an interesting job - "was to create the preinstall bundles, the drive images that were then squirted onto each new computer as it got to the end of the production line. The images contained..."

Leo: Eww, you left your drive bundle on my computer. Eww. Ick.

Steve: "The images contained an installed version of Windows, along with drivers and a bundle of games and other apps that came 'free' with the machine."

Leo: Yeah, right.

Steve: And think about the problem, though, of doing that. He said, "It was a fascinating job because different machine configs needed to have different bundles of software on them. I had to examine the way each game or app installed itself so that I could later 'install' the software automatically by just copying files and adding stuff to the Windows registry. Apps with copy protection systems were often very tricky. You'd start with two identical blank drives, install Windows on both, install the app on one of them, and then compare them to see what had changed. Once you knew which files went there and what changes the app made to the registry and other system files, you could then create a

batch file that would be able to silently and quickly drop a 'installed' version of the app on a new drive."

Leo: [Snoring]

Steve: Hey, Leo, somebody's got to do this.

Leo: Somebody's got to do it.

Steve: Somebody's got to do it. He says, "It felt like legalized cracking."

Leo: Yeah.

Steve: "In some cases it could take a couple of days to fully dissect even a single app like this, Lotus Notes" - can you imagine? - "being a prime example. And backing up the drives you were working on just wasn't practical. Some apps seem to know they've been copied to another drive, even if you use cloning software to supposedly copy the drive byte by byte. And of course we were throwing hard drives around the place all the time, plugging and unplugging them, reformatting, trying again, more plugging and unplugging.

"Many drives" - and here's his point - "bit the dust from the sheer physical demands we put on them, and did so without notice, until the day my boss Duncan bought a copy of SpinRite. Instead of trying to back drives up, we just SpinRited them all every few nights. SpinRite kept those drives going for ages. And, more importantly, it told us when the drives were getting a little too knackered, so we could take them out of commission before being tripped up by them. It was like night and day. We still threw the drives around with abandon. But SpinRite was like a guardian angel, restoring and protecting them as we went.

"You changed the way we worked, Steve, and saved us a huge amount of hassle and worry. The company is gone now, but no one in the R&D department left without having learned about SpinRite. Thanks, Dude."

Leo: That's awesome.

Steve: Signed, Howard. So thank you, Howard, for sharing that with everyone.

Leo: Yeah. Nice story. We're going to take a break. When we come back, we've got questions, Steve's got answers in Security Now! Episode 395. In another month we'll hit 400.

Steve: Coming up on, oh, 400. I like round numbers.

Leo: 400. Don't know what it means. I told - you know they just celebrated 100 episodes last night on All About Android. I said, you know, anything less than a few hundred we don't really - we've got shows in the thousands, kid.

Steve: Where are you [indiscernible]? You must be into the triple digits.

Leo: Oh, 1,400-plus, yeah, I think it's 1,409, something like that [The Giz Wiz]. But it was - that was because it was a daily show.

Steve: Right. Was?

Leo: And I'm sure TNT is - yeah, no, it's weekly now. We made it weekly.

Steve: Oh, okay.

Leo: Yeah. I bet you TNT is up to that.

Steve: I'll never be able to catch up, though, darn it.

Leo: No.

Steve: And we're not making us daily. That would kill me.

Leo: How would we get to 1,400? It'll be, like, years.

Steve: We could cheat. We could just, when we get 400, we could add a one on the front. No.

Leo: Questions for Mr. G. Are you ready?

Steve: Okay, so here's what I think, Leo. Because I am conscious of the time and that you've got another podcast to do, how about if we did half now?

Leo: Sure. Half next week?

Steve: And then we'll do the second half next week because I think the topic for next week, actually the last question of this week was going to be a lead-in about distributed hash tables. This is the directory system that Tor uses. I mentioned when we were talking about Tor last week that a directory, that services posted their existence in a

directory. And several people said, directory? How can you have a directory in Tor? So this uses - Tor uses a technology called "distributed hash tables" that we've never, ever spoken about, that I thought would make a great topic. And I think I can do that and the other five questions in one podcast next week. So let's do five today, and that'll give us a nice wrap. And then we'll do the next, we'll do the second half next week.

Leo: Sounds good to me. Starting with Nicholas Bauer, Atlanta, Georgia. He's been thinking about loops and parallels: Steve, this week when you explained - which was last week, I think - the ability to paralyze chained - I'm sorry, parallelize, very big difference between "paralyze" and "parallelize" - chained ciphers, I think you missed the key part that was confusing the questioner, which is that usually you aren't trying to compute just one answer, but thousands of them. The idea, which I think was lost on the questioner, is to implement something like an assembly line. As soon as Unit A has computed the output from Input 1 and passed it off to Unit B, Unit 1 can immediately work on Input 2.

Thus, if the hash we are trying to break is 1,000 chained iterations, let's say we can do one iteration per second, normally it'd be 1,000 seconds. If instead we implement in hardware 1,000 dedicated units, each of which computes one iteration and passes it on to the next unit, we can do 1,000 units simultaneously, so in other words a million hashes per second. And I think you understood that. I don't think that was what we were talking about. But thanks. Keep up the great material. It helped inspire me to do some hobby programming and be fully conscious of security while doing so.

Steve: The one thing that I liked about this, and the reason I saw it, was "assembly line." That's all anyone ever has to think.

Leo: Right.

Steve: That's the most perfect model of this notion of pipelining. I mean, that's what an assembly line is. It's a pipeline of work. And if you are ever trying to either conceptualize it for yourself or explain it to somebody else, that's it. The analogy is the assembly line. It's perfect.

Leo: Sunil Joshi, Chicago, Illinois, reminds us about OneID. Steve, I remember you talking about OneID some time ago. I see their site's still functional and OneID being offered. Is OneID an OpenID hosting site?

Steve: No.

Leo: What is it?

Steve: So, okay. These are guys who - there's a big team of people. I wish them no ill. But they are trying to do a proprietary replacement for website logon.

Leo: Oh, no, sorry.

Steve: And that's the problem. There's a serial entrepreneur, Steve Kirsch, whom I've met, who's been successful previously on some major endeavors. He did Frame, you may remember, that was a major publishing platform, and other successful things in the past. And he's decided he wants to solve the problem. Well, my hat's off to him. But it's proprietary. And so what this means is that, in order to implement it, it requires websites to support his solution. And only people who have his solution can use his solution. And I have - several times I've gone back, wondering how they're doing. And today, the most recent list of websites you can use is something called Portero, authentic pre-owned luxury, whatever that is, used luxury; NVX, don't know what that is. There's a site called ShoppersChoice.com under which is BBQGuys.com. Their slogan is "We smoke the competition."

Leo: I like that.

Steve: Ultimate Patio, CookwareCenter.com, Home Appliance Center, and SportsmanGuys.com. You can log on with OneID if that's where you want to go. EmbroiderMyStuff.com, Bikes for the World, the League of Women Voters, Earth Charter US, Children Can't Wait, and Bill Foster. And that's it.

Leo: Well, and there's an existing technology that is open and in widespread use called OpenID that really is the same thing; right? There's no reason to use some proprietary solution.

Steve: Well, no. Well, okay. So OpenID attempts to use the existing infrastructure to solve the problem. Steve's solution is better. I mean, OneID is like - it's better. It's very good crypto. I've asked them for documentation. They have never been able to share any. He sends me pictures of arrows pointing at things. But that doesn't do it.

Leo: I'm sorry, it's proprietary. I don't understand.

Steve: And that's the point, is that, if the whole world were already using it, and anyone could, then it would make sense.

Leo: You know, Microsoft tried this, and nobody did it. I don't know why somebody else should...

Steve: With Passport.

Leo: With Passport, yeah.

Steve: Yeah, I know. So anyway, that's where they stand. I don't - they're not getting

off the ground. I mean, this is where they've been. It's been more than a year. And the point is why would a site like Amazon or eBay want to add that when, first of all, you have a real chicken-and-egg problem, and it's going to confuse people? It's like, wait, wait, do you need some new something or other? I mean, it's just - I don't see the value to the site. So it's going to - it's a tough sell. I mean, again, I wish them luck. But I don't think that a proprietary solution to logging on should win. I don't think it will.

Leo: No market for it.

Steve: Because I - yeah.

Leo: Yeah. John Chybowski, Poughkeepsie, New York, is wondering: How future-proof are LastPass and similar services? Love the show, blah blah blah. What if LastPass goes out of business? My entire online life is tied up in this excellent service, but I'd still prefer not to have the company's wellbeing act as a single point of failure, and I'm not sure if LP is completely stand-alone. You still need to use their servers to maintain your personal blob of encrypted information; right? Of course I don't expect this to happen any time soon. But, hey, what happens 10, 20 years down the road? That's a good question.

Steve: It is. And I liked this because it reflects the fundamental tension of the cloud. I mean, for all the benefits it offers, with it comes some concern like this. LastPass in particular is very nice because the blob is encrypted and cached locally in all of the various platforms, services, machines where you use it.

Leo: Yeah, because I can go offline and still use LastPass.

Steve: Yes. So that's very cool. And the format is so clean and well designed and documented that they even have, at least for Windows, and I would imagine other platforms, I don't know for sure, a blob browser, where you can export your blob to your machine and view it in a standalone LastPass viewer.

Leo: Right.

Steve: Which is also very cool. So they've done everything they can to solve this problem for us, if they just, like, went up in a puff of smoke somehow. So the value they add is the centralized cloud interconnectivity, where if I change a password in Firefox on my Windows machine, and then I'm using Chrome over on a Mac the next day, it knows. It got the update. And so when I log in, it's synchronized. I mean, that's just very cool, the notion of cloud-based synch. So LastPass in particular I think has done everything they could to make it a non-emergency if they disappear. If they did, if they puffed a smoke, we're still able to log onto everything we can now. We would lose the inter-device synchronization which they provide. But it wouldn't be the end of life. And right now LastPass knows way more about my passwords than I do. I don't even try. It's just like, okay, that's - as long a site accepts the gibberish that I just gave it, and LastPass remembers it for me, everybody's happy. But it is a tension.

I would argue, though, that 20 years down the road, I mean, that's a long time. We didn't have the Internet 20 years ago. Zero. Didn't exist at all. So we're not going to have this problem in 20 years. I don't think we'll have it in 10 or five. There will...

Leo: There'll be a solution.

Steve: ...be a solution.

Leo: And that's one of the reasons I like LastPass is I give them a buck a month for the premium account. They have a business model. It just seems like they're going to be here long enough that we can hand it off to something else. I just recently moved, because of the Evernote thing, I moved all of my passports, Social Security numbers, all my personal documents into Secure Notes on LastPass. So now I really am in there.

Steve: You were right, also. There was a big dialogue in the GRC newsgroups about this following on our dialogue about it last week, Leo. And the consensus was LastPass had - I mean, I'm sorry, Evernote - this whole weird "We can't export more than 64 bits." There are a couple good crypto guys who hang out in the newsgroup; and they were talking about, you know, if they're using RC2 with 64 bits, eh, that's really not the crypto that you want. So your notion of getting it out of Evernote and putting it into LastPass really does make more sense. You still get the cloud synch, but you get really strong crypto.

And you've got to wonder what, I mean, I did some research after we spoke. And sure enough, in the Evernote docs they say, well, we don't have enough staff to go through the hassle of applying to the State Department, or Commerce, rather, to get the certificate to allow us to use strong crypto, so we're going to go with 64 bits. Uh, what?

Leo: Yeah, whatever.

Steve: Okay, my browser is using 128 when it talks outside the country.

Leo: I'm going to keep using Evernote. I love Evernote. But I'm not going to store anything secure in there anymore.

Steve: I think that makes perfect sense.

Leo: I'm going to use LastPass. And really, I tell you, I started using a Chromebook as for a review. And it's so great because while the Chromebook is just Chrome OS, it's just a browser, as soon as you install LastPass into it, you feel like you're right at home because I can immediately do everything I want to do. My data is there. So it really is nice to have that.

The only thing, and I'll have to look into this, is there is an open source alternative called KeePass, K-e-e-P-a-s-s. And it's conceivable that it would be better to go with

an open source alternative, if it had the functionality. I know it doesn't. But if it had all the functionality of LastPass...

Steve: It probably doesn't have the - does it have the cloud glue?

Leo: That's a good question. I don't know if...

Steve: That's a new term, by the way, Leo. We've got cloud glue now.

Leo: Cloud glue. I don't know about cloud glue.

Steve: He got cloud glue on his head, so he got it off.

Leo: I'll have to ask about cloud glue. I'm not sure how they do the synchronization.

Steve: Oh, that's the cloud glue. We don't know about that.

Leo: I'm not sure how they do that. It may be that what you do, and I suspect this is what you do, is you use something like Dropbox to store the encrypted blob and then have that synchronized to all your machines.

Steve: Then you're cloud vendor neutral for the glue.

Leo: Right. It's just a blob.

Steve: Yeah. Of glue.

Leo: So that's an interesting idea. So, yeah. So everybody in the chatroom is agreeing that that's exactly it, is you have this blob, and you're responsible for gluing it. In other words, use something like Dropbox or ShareFile or some form of sharing it with the rest of your machines.

Steve: And otherwise the functionality is similar? It's cross-browser, cross-platform, fills in the forms, offers passwords, blah blah blah?

Leo: It's missing some extensions. There's no Firefox extension.

Steve: Uh, thank you very much, then.

Leo: Yeah. That's the beauty of LastPass is it works on every platform, automatically, kind of.

Steve: And they are a business. Their motives are clear. They've withstood attacks. They've acted responsibly. They're full disclosure of all of their technology. Joe's, I mean, Joe's present and working in the forums. And because there is, you know, we're paying them a buck a month, there's money there for them to add support as new platforms come out, you know, like on my BlackBerry.

Leo: I'm always nervous when a company like this, that I'm relying on, doesn't have a business model because that means they may not be around for a few years.

Steve: Yeah.

Leo: Daryl in Kansas reminds us of the wacky Mailinator.com service: Steve, I think this is an excellent tool to put in your toolkit. I thought listeners would like it, too. The geek factor is high. Here's what the website says about Mailinator.com: Use any inbox you like. No sign-up. Inboxes are created when email arrives for them. Make up email addresses on the fly. Make up addresses, give it to somebody else, come here, check the inbox. Atom feeds for every inbox so you can push/pull. Give out a Mailinator address anytime you need an email address, but don't want to get spammed. And then he says thanks for SpinRite and ShieldsUP!. Hey, Leo. Mailinator. So it's like a way to create dummy email addresses.

Steve: Well, no. Well, okay. First of all, we did speak about this once. And we haven't talked about it for a long time. It's just so loony that it's fun.

Leo: It just might work.

Steve: I would never have come up with this because it's so wrong. But there's a purpose. And that is, so here's Mailinator.com. And it will accept email sent to it from any account incoming. So an SMTP server connects to it and says I've got mail for LeoandSteve at Mailinator.com. And Mailinator says, good, send it. And Mailinator accepts it. And that's what it does. You don't have to - there doesn't have to be a SteveandLeo Mailinator.com account. It accepts any mail for any account, anything, and just puts it in a database. And then you can go to Mailinator.com, and right on the front page is a form, a little field. You put in SteveandLeo and click "Check my mail," and it shows it to you. There's no security. There's no account. There's, I mean, that's what it does.

Now, the mail lasts, oh, I think he's - I think it was overnight, or maybe it was a week. I don't remember how long it was. But so it's not guaranteed to be there forever. And not only can you RSS or Atom feed, now they support POP. So you could have your own web client polling Mailinator.com once a minute or so for a certain mailbox at Mailinator.com, SteveandLeo.com. And anything that arrived there, your client would pick up. Again, no security. So...

Leo: Anybody can read that mail, but that's fine. Just don't have passwords sent there.

Steve: Yes. And that's the key is - I had forgotten about it. And so for certain things it makes so much sense. If you have an absolute throwaway confirmation link for some random place that is annoying you, that you've got to prove who you are, you can just type in qwerty@ -- well, actually that one's probably taken. Actually, it's kind of fun, Leo. Go to Mailinator.com and put in - I put in "Noodles," and there was no Noodles. But then I put in "Charlie," and Charlie's got a bunch of mail. And so you could just put - try "Monkey." I wonder what Monkey's got.

Leo: Oh, Monkey's got a ton of mail, I'm sure. So really this would be mail as only one thing, mail that came to Charlie@mailinator.com. You can, by the way, it will accept mail from any - so you could actually point your domain, what would be the MX record, to Mailinator. And so you could create your own email address that doesn't say "Mailinator" and still use it as a throwaway. But the key, again, is do not use this for anything that would, for instance, be a password recovery because, well, you're just - that would be dumb.

Steve: Right. It is. So it is - what it is, for what it is, is I think wacky. I mean, it's wonderful. It's crazy.

Leo: It's just for using as a throwaway address where you know it's just going to be spammed, basically.

Steve: Yes. And, well, and anybody can see what was sent there. So the normal purpose of email, an example you gave, password recovery, is that the assumption is only you can access your email. Therefore something very critical, I mean, it's like the only - email, an email loop, except for now we have cell phone usage that is so ubiquitous that that's now becoming a solution. When I logged into Google Docs just now, earlier, to produce the docs, Google wanted just an update on my authentication, so it sent me a six-digit code to my phone. And it came through the loop, I typed it in, and Google was happy. So that's nice. We have that now.

But email is still what most people use. And it's because only we are able to access our email account, hopefully, that you're able to use it for this kind of thing. So Mailinator is not that. It's just, you know, it's just crazy that you just make up a mailbox and send stuff there, and it's there for a while, for like a day, and then it just kind of expires off the other end.

Leo: You've been Mailinated.

Steve: I think it's very cool. And now you want to jump to No. 10, which will lead us into next week.

Leo: Excellent. Spencer, writing from the canonical "undisclosed location," poses two questions about Tor v2.0.

Steve: And now we know why his location was undisclosed.

Leo: Uh, Steve, you do a great show on TWiT, but I've got questions. One, in your explanation of the old version TOR, you never told us how the return packets make their way back to the anonymous user. How does that happen? We do have a whole show on that subject, by the way [SN-070]. Two, you said these new 'Tor services' are published in the 'Tor directory' so users can find them. Well, where is this directory, and wouldn't attacking it bring the whole service system down?

Steve: Okay. So first one is that Tor nodes do record the information necessary to locally return an incoming packet to where it came from. So in the same way that a node knows who sent it and where it is being sent to, and it knows its own crypto, if you think about it, as long as it retains some short-term memory of that information, when the returning packet arrives from the place it's sent the packet out to, it identifies what that's a connection from, sort of in the same way that NAT routers work.

We've talked about NAT routing tables where the act of the packet leaving the NAT router from one of many machines behind it allows the NAT router, when the packet comes back from the outside, to determine which machine should receive it. So it's very much the same way, just a simple short-term routing table that allows it to do that. And it also then reencrypts the packet with its private key before it sends it to the place where it received it from. And so the chain is reversed in a similar fashion. Then it gets back to the user, who has exactly the same kind of onion that they sent. And since they know the sequence of machines it came through, they sequentially decrypt each round or each layer of the onion with each node's public key and get the returned data. So that's No. 1, just basically backtracking and just remembering sort of some routing information the way a NAT router does, with the addition of the crypto.

And the Tor services published in the Tor directory, this is done, as I mentioned at the top of the show, and this will be the subject for the first half of next week's podcast. We will wrap up next week's podcast with the five questions we didn't get to today. And that is the technology of distributed hashing tables. The key concept, which is cool, is that you locate something by its hash. And that's all I'll say for now.

Leo: I can't wait.

Steve: And we'll talk about it in detail. You know we will. Get your propellers ready.

Leo: [Laughing] I love this show. Steve Gibson, he is our Explainer in Chief. The most geeky show on the network, and I'm mighty proud of that. You can listen to Security Now! every, what is this, Wednesday, 11:00 a.m. to - what day? Where am I? Who am I? What am I doing here? 11:00 a.m. Pacific, 2:00 p.m. Eastern time, 1800 UTC. Note the time change. We're in summertime now here in the U.S. So that means UTC, which doesn't adjust for summertime, maybe someday we won't either,

but for...

Steve: That means you were actually early for the podcast, Leo.

Leo: UTC I was.

Steve: Yeah.

Leo: Don't - yeah. Yeah, well, now what do you say? I'm going to - somebody bit my head off. Anyway, the best thing to do is watch live, if you can. But if you can't, don't worry, we've got audio and video available on demand after the fact at TWiT.tv/sn. You know Steve has 16Kb audio versions for those of you who really have bandwidth caps, or even text versions. He pays for transcriptions, human transcriptions, written by Elaine, and you can get those at GRC.com. While you're there you might want to check out SpinRite, world's finest hard drive maintenance and recovery utility. If you've got a hard drive, you've got to have SpinRite. You can also get all the free stuff Steve makes available to us, including the UPnP tester on ShieldsUP!, which you really ought to...

Steve: Oh, and that count keeps going up, too. I saw it the other day. I think we're like, 3,200 something. I think that's what it was last.

Leo: Holy cow. That's 3,200 routers who are prey to the UPnP exploit from outside the router.

Steve: Yup. I did receive a note from somebody who had two D-Link routers hacked from the outside. And so one had some sort of trojan installed in it that had taken it over, and it was being used as a trojan relay. And the other one had all of its ports opened up, so it was wide open to the Internet. Yikes.

Leo: Wow. Wow. ShieldsUP! Run it. GRC.com and check out all the other great stuff there, including the feedback section, GRC.com/feedback, if you want to ask a question for future episodes. We'll save those other five, do them next week or the week - whenever we get around to it.

Steve: Yeah.

Leo: Yeah. Thank you, Steve.

Steve: Leo, always a pleasure. Talk to you next week.