# SECURITY NOW!

Transcript of Episode #302

## Listener Feedback #118

**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-302.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-302-lq.mp3

**Leo Laporte:** This is Security Now! with Steve Gibson, Episode 302, recorded May 25, 2011: Your questions, Steve's answers, #118.

It's time for Security Now!, the show that protects you and your loved ones online, your privacy too. And here's the guy who does it all, from GRC.com, the great Steve Gibson. Hi, Steve.

**Steve Gibson:** Hello, Leo. It's great to be with you again, as always.

**Leo:** Let's get ready to secure yourselves.

**Steve:** This is Episode 302.

**Leo:** Oh, my god, I can't believe it. You know, you're just one behind, well, I guess you're, yeah, you're just half a week behind TWiT. Because TWiT did 302 on Sunday. I know that was your goal.

**Steve:** And TWiT started before Security Now! did. You had them as your flagship podcast. But since we keep counting ours, and you've had some TWiT vacations...

**Leo:** Yeah, but never again, by the way. Lisa has informed me that I may no longer miss any episodes of any show ever, ever again. No, TWiT especially is a show we don't want to miss. So unless something surprising happens, you're going to be at parity forever.

**Steve:** Sounds good.

**Leo:** I think that's fine.

**Steve:** Yeah.

**Leo:** I think that's fine. You caught up. That was amazing.

**Steve:** So we have our regular roundup. This is a Q&A episode, our 118th Q&A. We've got a roundup of News of the Week and some great comments and feedback from our listeners to cover this week. And the last question is actually a lead-in to next week's episode that is going to surprise everyone.

**Leo:** Oh, really. Something new. Something exciting. Something different. Something we've long awaited?

**Steve:** I would call it, actually, no, it's probably nothing less than a breakthrough.

**Leo:** Whoa.

**Steve:** So, yeah.

**Leo:** Whoa.

**Steve:** Yeah. It's going to...

**Leo:** All right. Whoa.

**Steve:** Yeah.

**Leo:** Before we get to it, I'm just looking at the notes, we've got security updates to talk about. There's a lot of security news. Sony's been breached again.

**Steve:** Oh, they are just - they ought to just get off the Internet.

**Leo:** Well, now it's like a matter of pride, I'm sure, for hackers. It's going to be really tough for them to get off this horse.

**Steve:** Well, we're going to talk about what happened because it's something we've talked about before, which is, well, I won't...

**Leo:** Save it, save it.

**Steve:** ...negotiate with myself. Yeah, I'll save it.

**Leo:** All right, Steve. Time to get to the security news.

**Steve:** So Google has been updating Android, actually 99.7 percent of their devices, of the Android devices. SANS wrote that "Google is rolling out a fix for a vulnerability in the majority of Android phones that allows attackers to access and modify users' Google Contacts and Calendar when they are being accessed" - and listen to this carefully because this is important for us - "when they are being accessed over unsecured WiFi networks."

**Leo:** It's a Firesheep thing.

**Steve:** Exactly.

**Leo:** Now, didn't they fix it server-side, though, right away?

**Steve:** Well, the flaw, for those who are interested, the flaw affects versions 2.3.3 and earlier on the Android platform, which is virtually all, all but .3 percent of Android devices. And it requires no action from the users. It'll be pushed out automatically. The reason this caught my eye is that we've had a number of listeners who have essentially asked this question, which is, well, if I'm using a smartphone which has both a 3G cellular connection or a WiFi connection, but as is the case with smartphones it will preferentially use WiFi when it's available, what do we know about the security of the phone's use of WiFi versus 3G?

And this is a perfect example of the concern that this creates, which is, if you're not using your own WiFi, which we know would be secured, and if you're using open WiFi, that is less secure from in the air near the phone than if you were using 3G, which, while its cryptography is not uncrackable, it's still very strong and much better than, as you say, the Firesheep stuff, which basically just takes advantage of the fact that there is, over an unencrypted connection, there is data moving between the phone and the access point which can be sniffed and is in the clear. And this was happening to people who were using Android phones.

So the question that's been posed to me, I've seen it a number of times, is should I turn off my WiFi when my phone would like to use it, but it's unsecured? And from a security standpoint, I would say yes. I would say, unless you really need the bandwidth, if you're just doing things that can work over 3G, if the choice is between that and open WiFi, I would use the cellular connection because it at least has local encryption. And so, while Google has fixed this for Contacts and Calendar, we don't know what's going on with all those other apps that are loaded on the phone that might very well have...

**Leo:** That's a good point, yeah.

**Steve:** ...yeah, Firesheep-ish problems. I mean, it's unfortunate...

**Leo:** Like the Facebook app, for instance, or something like that.

**Steve:** Precisely. All of the other apps that are doing communication, they're probably glad to be on Android. But security may well not be a priority for them.

**Leo:** If they offer SSL, as some do - like Foursquare or whatever, Twitter, do that; right?

**Steve:** Oh, absolutely. So, for example, in the case of Facebook, if you're able to turn that on and keep it on - of course we know the problem with Facebook is that there are still Facebook apps which don't support SSL, which unfortunately require that you drop completely out of SSL rather than it being more granular. I'm hoping that Facebook is first going to make it more granular and then go the next step and just make it a requirement of apps to bring their security up, if they want to play in Facebox world. Facebook's world.

And Adobe has one again updated Flash. I noted that they're no longer even talking about quarterly updates.

**Leo:** Not even close.

**Steve:** They've dropped that whole idea of, well, we're going to wait till June. Fine, just give it up. Because they had a big problem which was affecting all versions of Flash Player for Windows, Mac, Linux, and Solaris, any versions 10.2.159.1 and earlier. And the same thing, 154.28 for Chrome, 157.51 for Android. And I just, when I turned my Win7 box on in order to fire up Skype for this conversation with you, Leo, I got the notice, and what they were giving me was 10.3. So I imagine that's where they moved to. So you do want to make sure that you're using the latest since it's broad platform.

And so they've released patches for multiple security vulnerabilities affecting Flash Player. And Adobe reported that malware in the wild is exploiting at least one of those known vulnerabilities, which is a memory corruption problem. And so merely by enticing a target to view a maliciously crafted file, an attacker can exploit at least one of those vulnerabilities, well, actually all of those, but has been exploiting one of them in order to execute arbitrary code. And so I just see Adobe saying, well, we're just going to - we're going to update our stuff as we need to and not try to follow any schedule because that hasn't worked very well.

I did want to mention, since we're talking about Facebook, that they have added phone-loop, two-factor authentication. Some users have been a little put off by that because it requires giving Facebook your cell phone number. And people that have been concerned about Facebook's privacy practices are already sort of fed up with telling Facebook anything that they're worried Facebook may leak. But the two-factor authentication approach seems to be a good one. They probably use cookies because they identify machines that you have used before.

So the idea is, if you attempt to log on, in their jargon, from a device that they have not seen you use before, then and only then, if you have it configured, and it's not automatically turned on, so Facebook users need to go and activate this in order to make it happen, you activate it in your security settings, give them your cell phone number, they will send you an authentication code which you then need to enter into your logon point on this device where you've never done it before. Then they stick a permanent cookie on that device which authenticates it in the future. Which is a really nice tradeoff. It means you're not being harassed all the time. I think it's typical Facebook user-friendly.

I got a tweet from someone saying, well, yeah, but why aren't they supporting VeriSign or YubiKey or blah blah blah? It's like, well, I would call that advanced one-time password multifactor authentication. It's probably outside the scope of what most Facebook users can do. But for people who know that they're going to have a cell phone around, at least when they're using a new device, this is great because it means that the bad guys aren't going to be able to close that phone loop by providing the information. Also, you'll know if someone is trying to log on as you because your phone will send you a code that you're not expecting. Which means someone induced Facebook to send you this. So it's also sort of a cool notification that someone is trying to get into your account. So I think it's a good thing. It's certainly a step forward. And we could say long overdue. But it's great that it's here finally.

And Kaspersky has found operating in Brazil, and it's just sort of located there so far, another rootkit, another Windows rootkit which boasts x64, that is, 64-bit support. It's a variant of a prior rootkit known as the "Banker Rootkit," which targets online banking access credentials. Okay, now, this ties into a question that we're going to run across later on. Of course, I guess everything we talk about ties into everything else. But this thing leverages a hole in an obsolete version of Java.

So we've talked about the problems with just pretty much having anything on your system that you're not using. And I have in the past encouraged users, if they don't know they need Java, to remove it. Sun is doing as good a job as they can of keeping it current. But they're only able to patch the things they know about. And it's the things they don't know about that can catch us out. So if in this case users are not keeping their Java current, but something installed it on their Windows machines in the past, and they do whatever behavior it is, which wasn't made clear in Kaspersky's note, that they visit a website that leverages Java to install this.

The first thing this thing does is it disables Windows User Account Control (UAC) because that just gets in its way. And that lets it go about its business without being interrupted because that User Account Control, that's pesky, you know. You don't want that if you're malware. Then it installs bogus root certificates into the system and modifies the hosts file so that accesses to popular banking websites are redirected to phishing sites operated by the criminals. Thanks to the fact that this thing installs root certificates in the user's machine, no SSL warnings come up when the user is connected to the wrong system.

The hosts file is like a DNS patch. Remember that Windows systems, well, actually all UNIX and Internet-connected machines have a hosts file, as do Windows and Macs and so forth, where it looks first, before it does any other DNS lookups. So they put the domain names of banking sites, the legitimate domain names of banking sites into the hosts file, associating them with the IP address of their malicious server. When the user puts correctly, like clicks on a valid shortcut that they had set up beforehand, a true and correct URL, their system looks at the hosts file for BofA.com or Brasilia, whatever, gets the wrong IP because it is no longer using DNS.

The hosts file hack intercepts DNS. Their browser then connects to the malicious IP. Over SSL, that's the other thing. The padlock is there. SSL is shown. I mean, everything about this looks legitimate because this rootkit has installed a bogus certificate which the malicious server has had its certificates signed by. So SSL, proper URL in the browser, nothing looks wrong about this, yet you are connected to a malicious site where you're then going to provide your logon credentials. And basically that allows this rootkit to obtain your banking access credentials by spoofing the proper appearance of the web page. Nothing that you see looks out of order.

**Leo:** I like it that it's called the "Banker Rootkit."

**Steve:** Banker Rootkit, yeah, because it's going to suck out all your money.

**Leo:** Now, Bank of America does a SiteKey, for instance.

**Steve:** Yeah, and this defeats that, too.

**Leo:** Doesn't help.

**Steve:** Right, because what it does is it turns around and obtains the page, exactly.

**Leo:** That's what I always wondered about SiteKey, is like, how it this supposed to help? If I can see it, a bad guy can see it.

**Steve:** This essentially is a man-in-the-middle attack where, thanks to the fact that your local system has been so deeply compromised with DNS redirection thanks to the hosts file, and bogus root certificates thanks to the fact that this rootkit was able to bypass UAC. The other thing it does that you just sort of shake your head at is it needs, in order to pull off some of this, it needs to install a kernel driver. Well, we know that 64-bit systems are protected with PatchGuard that we talked about years ago when Microsoft introduced this in Vista. So now it ought to be very mature. Now we're at Windows 7. And PatchGuard, one of the things it does is absolutely require that drivers be signed. Except, you know, Leo, that's inconvenient for developers because when they're writing drivers and compiling them, they don't want to have to keep signing them to test them because that's pesky. So there's a "test mode," which bypasses that requirement.

**Leo:** And it's accessible to anybody, of course.

**Steve:** Of course. And the rootkit says, well, having bypassed the User Account Control because we don't want those pesky little pop-ups to warn the individual that we're modifying their system, we'll turn on test mode so that we don't have to bother with having signed drivers because we don't have any.

**Leo:** What a pain.

**Steve:** Yeah.

**Leo:** Now, if I have - okay, one more question. The chatroom is now all atwitter. I have on my Bank of America accounts and my PayPal account and so forth turned on that cell phone second-factor authentication; right? So I log in, and it says, okay, send this to your cell phone, and I enter it in. I guess because it's on my machine, it's still there. I mean, once I get into BofA, I mean...

**Steve:** They would be able to hijack that session, but they would not be able to hijack successive sessions.

**Leo:** They have to do it separately each time, of course.

**Steve:** So that logon they would be able to hijack because the bank would send you the authentication, and you would go, oh, look, I mean, that would almost prove to you that you're talking to your bank because you would get the second factor or third factor, whatever factor it is, challenge, which you would respond to, typing it into the page. The bad guys would forward that, because they're intercepting your traffic, to the bank. The bank would be happy, and you'd be able to log on. They could not, however, reuse your credentials in the future.

And if they were smart enough, and if they're not now, they will be, to recognize that you're using a multifactor authentication, they would do all their dastardly work right then and there. And they have a connection with your bank. So as soon as you've given them credentials access, right then they're able to do what they want to. And we have seen banking trojans which will empty, which transfer all of the money out of your account prior to you being able to complete the first transaction of the session. So the bad guys understand this. The moment they get access, they turn around and send all your money off to Russia or wherever. And you look at your account, and it's empty. So they don't even need to come back for further access. Although that's one of the things this thing apparently does acquire.

**Leo:** Does it impersonate your computer, if there's a cookie on my computer that says I'm me? Because that's another thing banks do.

**Steve:** Oh, absolutely, because remember the cookie goes through the channel. And these guys get - they're monitoring the channel. They have full man-in-the-middle activity. So even if the bank said you have to do special things if you log in from a computer you haven't used before, your computer sends that to the bad guys, who then turn around and send it on as part of - just it's echoing the traffic that your computer sends. And so here we're looking at the future. And it's a rootkit, meaning that what that driver has done that they've installed is it's patched the actual API, the low-level operating system function, so that you can't see any of it. It's removed itself from the directory listings. And so things that scan your system, which are unable to scan underneath the rootkit, which is an additional layer of difficulty, and rootkits are becoming increasingly aware now of anti-malware and anti-rootkit technology, these things are becoming insidious. And, I mean, this is a perfect example of, once something like this gets on your system, you've pretty much lost your system. You just need to start over.

**Leo:** And what's the attack vector?

**Steve:** Java. If you didn't have - I know. I know. So you go to some site that is malicious, or that the bad guys have installed using, for example, an SQL injection attack. So you're just - you're at DSL Reports, because they had a problem recently with SQL injection. And you're just looking at a forum. And you trust DSL Reports because it's a good site. But they've got a problem with their backend server technology. You browse someone's posting. Just looking at it in your browser allows this exploit to leverage the fact that you've got a known problem in the version of Java on your system, and the jig is up.

**Leo:** Oh, boy.

**Steve:** Yeah, I mean, this sounds like science fiction, but this is what's happening now.

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

**Steve:** So I did want to follow up a little bit on - apparently this is just in Brazil. So it's not widespread yet. Kaspersky found it. They got a copy of it. They reverse-engineered it. They figured out what it was doing so that I was able to tell our listeners. But there's just no better reason to keep yourself current, patched, with the things that we talk about, and run with the minimum target profile, which frankly means not having JavaScript on because it's JavaScript that then allows Java. JavaScript invokes Java in order to do Java applet sorts of things. So you really, really want to just, where you can, minimize the profile that you're presenting to the target.

Speaking of which, there was a little bit more, we talked about it last week, sort of the questionable conduct of Apple support for their AppleCare customers. Now we've had a support document apparently leaked from Apple which directs AppleCare workers not to, quote, "confirm or deny" whether a user's Macintosh is infected, and not to attempt to remove or uninstall any infections. At this point, I mean, I give Apple some leeway because frankly they're late to the game in terms of how you handle these kinds of problems. They've had the privilege, the benefit for so long of having a small enough market share that they just weren't a target for the attackers. And their prices were so high that Mom and Dad weren't buying Macs for Junior Hacker, they were getting him a

PC.

And as we've talked about before, hackers cannot develop attacks for machines they don't have. You can only develop attacks for the machines you own. And increasingly now those are Macs. Graham Cluley, who is one of the chief technical guys at Sophos - who is, by the way, as we know, Astaro's new parent - he wrote that Mac's increased market share has "Effectively ... reached a tipping point where people are now getting hit with malware on their Macs. On the support forums you'll see plenty of people who say they were just Googling around when a message popped up and convinced them they had a security problem." He says, "In terms of Mac malware, Mac Defender is the biggest event to date. There were earlier viruses and malware, but this one is big." So the Mac grows up, unfortunately, in a way that we've been discussing Windows' trials and travails for many years now.

**Leo:** And they're sitting ducks because Mac users haven't had to deal with this.

**Steve:** And that's a brilliant point, yes, we talked about that last week also. You're right, Leo, it's that in general there isn't a level of on-guardness on the side of Mac users because, as you say, this hasn't been the way their world has been until now. So, yeah. And I was about to say, again, that the problem is all of these systems are porous. And then my eye falls on the next topic here, under Attacks - speaking of porous. Under Attacks & Breaches is Sony's continuing trauma. Now that the target has been painted on them, attackers have just been having a field day.

Sony, I wrote "GMG," but that's GMC in Greece, and also some Asian sites, have been attacked. In the case of the site in Greece, email address, usernames, passwords, phone numbers, personal information, basically it's another big mistake on Sony's part. And this is sort of like, I mean, I liken this to the problems that Adobe had with Flash, is for the longest time Adobe was just pouring technology out, and nobody was trying to attack it, so Adobe didn't need to harden it against attack. And what we have as a consequence is a huge code base that has never been hardened, and it's huge, meaning that it's full of problems. Similarly, here we have Sony. Someone says, well, why are there so many problems with them? Well, the problems have always been there. But they never needed to harden themselves against attack.

And it's difficult to do. I mean, it's not the default case. The default case is something works, and so you put it online. And, I mean, we're talking about a world now that is different than it's going to be in a decade because the world certainly is quickly coming up to speed. And frankly, even in the United States, Congress is becoming increasingly concerned that super popular sites like Facebook that have tens, hundreds of millions of users are increasingly posing a threat to their own users.

So NASA similarly confirmed that an FTP server at their Goddard Flight Center had been breached in this last week. And the same person, a Romanian hacker known as TinKode, T-i-n-K-o-d-e, also breached the security at a European Space Agency network last April, last month. He refused requests to discuss the details of the network vulnerability he exploited in the NASA intrusion. Oh, and he said he had not been contacted yet by NASA. And he said that he had obtained confidential satellite data from that Goddard Space Flight Center FTP server. So our systems, unfortunately, are not secure enough to withstand really strong scrutiny when bad guys decide that's where they want to focus.

In miscellaneous notes, I wanted to mention that I've been getting great feedback from our recommendation of Mark Russinovich's "Zero Day" novel. Lots of people have purchased it. I got feedback quickly from people who bought the eBook version on the Kindle, saying that they were only into the first chapter, and already they couldn't put it down. So I just wanted to thank them for letting me know and to remind people that it's a good book.

And in our Unintended Consequences sideline, don't know if you saw this, Leo, but it turns out that Bitcoin Miners, that is, those who run high-power machines...

**Leo:** Like you.

**Steve:** ...like me, trying to mint bitcoins - I turned mine off because I figured I just got lucky making 50 bitcoins in less than a week.

**Leo:** Well, how do you get bitcoins if you don't make them?

**Steve:** Oh, you're able to trade them. You're able to buy and sell goods using bitcoin as a currency. But turns out that - initially these were just rumors on some IRC chats. But there has since been confirmation that bitcoin miners have been subject, in Canada at least, to some house searches. The Canadian town of Mission, BC has a bylaw on its books that allows the town's Public Safety Inspection Team to search people's homes for what's called "marijuana grow-ups" if they use more than 93 kWh of electricity per day.

So what's happening is, apparently, I guess a marijuana grow-up is you buy a whole huge, like, rooms full of grow lights, like fluorescent grow lights, in order to, like, have this incredibly bright, UV-rich environment to grow your marijuana plants. And it's detectable by the fact that you have a sudden spike in electricity usage. Unfortunately, that's the same characteristic as bitcoin miners.

**Leo:** Yeah, but you're not going to use 93 kWh a day, are you?

**Steve:** Apparently yes. I read some of the dialogue from people...

**Leo:** People must be having, like, 10 servers running.

**Steve:** No, they have a roomful.

**Leo:** Oh, please.

**Steve:** These people have gone nuts.

**Leo:** You know how much you're spending to make bitcoin?

**Steve:** And that's just it. Now, it was the case when we first talked about this that there was no economic - people were arguing that there was a power consumption cost disincentive. But Leo, I looked today at the leading exchange that is exchanging Bitcoin. It is north of $7, hovering around $7.50 U.S. per bitcoin.

**Leo:** So you made 350 bucks like that.

**Steve:** When you and I talked about it, when we first talked about this, it was like an event when it reached parity with the U.S. dollar, when one bitcoin was one U.S. dollar. We're now at $7.50. So that tilts, again, tilts the game in favor of bitcoin mining. Now, one of the other things that's happened, though, remember that one of the cool things about the bitcoin network is that it is entirely self-regulating. As this distributed peer-to-peer network sees that the rate of bitcoins being minted by these crazy people with houses full of servers and probably air conditioning, or at least their windows open and fans, it automatically increases the level of difficulty of the problem that must be solved, that is, this hashing problem that must be solved, in order to continuously regulate the rate at which new bitcoins are being minted.

So what this ultimately means is that it's going to be more difficult to mint bitcoins. And we know that the whole process is asymptotic. It's slowing down and leveling off over time so that there is only ever going to be, I think it's 21 million. I don't remember the number now. But some fixed number of bitcoins ever minted, like within X number of years. Again, I forgot a lot of these details. I knew them all...

**Leo:** To avoid inflation.

**Steve:** Exactly. You cannot inflate this. There is no way to create more bitcoins than the technology was originally set up to create. Now, I have to mention also here that I got a ton of tweets from our listeners who apparently picked up Jason Calacanis...

**Leo:** Oh, yeah. His link bait.

**Steve:** ...being quoted that "Bitcoin is the worst idea ever." And I've never watched Jason Calacanis. So I thought, what? What is he saying? Well, I still can't say that I have watched him. I tried.

**Leo:** It's just link bait. It doesn't even make any sense. I don't even understand what he's saying.

**Steve:** He's not very impressive. He's not a techno guy much. To all the people who tweeted me, I'll say that I wasn't endorsing Bitcoin, I was just - we were Security Now!. We were being what this podcast is, which is how does this stuff work? Is the crypto sound? Does it make sense? Can it be hacked? And the answer is this thing is really cool. It is a robust cyber currency that, as far as I can see, was absolutely done right. So that's all I'm saying. Jason must be grumbling that it can be used for terrorists to exchange value, I mean, who knows what it is he was...

**Leo:** I don't - I couldn't figure it out, what it was. I think it's just pure link bait. Rob Tercek responded, I think, with a pretty good rebuttal.

**Steve:** And frankly, I did watch those guys. He had two Bitcoin gurus on, and they were really good, articulate, and nice. And so if anyone's interested, if you want to find that - I don't know even where it was. I must have just put in "Jason Calacanis Bitcoin" or something.

**Leo:** Yeah, you can find it that way.

**Steve:** To find it on YouTube. So anyway, I don't know what he's talking about, but I'm not evaluating the morality of currency. The fact is, yes, you can use it for laundering, money laundering. It's not the money's fault, it's what you do with it. In the same way that you can use crypto to hide secrets that our states should have access to, but the bad guys use crypto. Well, it's not crypto's fault, it's the technology.

**Leo:** It's just a nongovernmental currency, that's it, right, in a nutshell.

**Steve:** Yes. And governments have a way of stomping those out.

**Leo:** Oh, yeah.

**Steve:** They're not happy with those. So in the Really Annoying Overreach department, we have the fact that the RIAA, our friends in the entertainment industry - who I was reminded have basically sued every technological advance that has ever been. I mean, they actually sued the people who made player pianos in the beginning because they felt that it was copyright infringement to have a player piano. And of course they famously tried to prevent the VCR from being made available to consumers. So everything that has come out which has ended up benefitting them tremendously, they tried to keep from happening. Now they have filed a legal action against Box.net, which is one of the leading cloud data storage providers. They are suing the cloud to get access to what users are storing in the cloud to see whether it violates copyrights.

**Leo:** Well, I want to have this case. This is good. We should get this court case.

**Steve:** Yes. There was an article, I'm reading from it, in Techdirt: "The RIAA really just doesn't know when to give up attacking and to start innovating. Its latest legal move is to file for a subpoena to get information from cloud storage provider Box.net to see if some people are using the service to store and share unauthorized music." Then I went back to the source article that Techdirt came from. And this was in Hollywood Reporter, reported that the Recording Industry Association of America, "fresh off a proclaimed 'milestone' in securing" their $105 million settlement against LimeWire, has now "set its sights on the burgeoning cloud-computing world.

"On Wednesday, the RIAA filed legal action against Box.net, a service that purports to let its users share, manage and access business content. The trade group seeks to investigate a couple of the company's users believed to be using the service to infringe sound recordings." And this goes on. I have the links in our notes. So I just wanted to point out that that's another reason why the concept of absolutely never putting anything in the cloud that you don't encrypt first one way or another, what I called "Pre-Egression Encryption," really has got to be the way we deal with cloud computing.

We still have the problem, of course, with cloud computing, of loss of access. So there are problems with putting things in the cloud. But people talk about security of the cloud all the time. And to my way of thinking, that's easily solved. I mean, you have to deliberately do it. You have to know what technology you're using and whether in fact it is doing Pre-Egression Encryption or not, that is, is it leaving your system encrypted with a key that the provider does not have so that it doesn't matter whether the RIAA or anybody else serves them with a subpoena, or law enforcement or anybody. They can say, well, here's the data, but we don't have the key. Or we used our key, and it still looks like gibberish. So good luck to you. Ugh. Anyway.

Oh, and from the Twitterverse, a listener, Michael Leonard in San Diego, he said, "Just listened to SN, great show. I think I have a better name for you: PIE, Pre-Internet Encryption."

**Leo:** Instead of PEE.

**Steve:** Which, you know, is urination, unfortunately.

**Leo:** PIE.

**Steve:** So I like PIE. And it's a little broader. And besides, "egression," it's like, okay, a lot of people don't know what that means, to egress.

**Leo:** I like PIE.

**Steve:** I do, too. PIE, Pre-Internet Encryption. It's easier, friendlier, and so that's the acronym from now on.

**Leo:** Tastier.

**Steve:** They're definitely tastier. And I did want to share a nice testimonial from Mark Wright in Oregon, a listener and SpinRite user. He said, "After having SpinRite for almost a year now, and using it on many of my systems, I finally have a wonderful testimonial. I frequently use SpinRite whenever I have strange problems with my systems. But until now, I have never actually seen it declare that it had fixed anything. Even though numerous times SpinRite reported no errors, the systems would then behave themselves from that point on."

Which of course is the case. We talked about that before. SpinRite often fixes things working with the drive, but doesn't report an error because it's reporting what it left behind, not what it encountered. And so it is fixing things, it's just there's no way really for me to say we fixed something because it's sort of the nature of the way it interacts with the drive. Anyway, he says, "A couple of months ago I gave one of my older PCs to a co-worker who couldn't afford to buy one. They called last week with that dreaded question: 'My PC isn't working. Can you fix it?'

"Well, the problem they were having was that Microsoft was now sure that they were using pirated Windows software, and they were getting all the warning pop-ups in XP for that. On top of it, they couldn't use the web because IE kept hanging and crashing. Since it was an older box, I decided to run SpinRite, and just let it run overnight. Well, the next morning, 12 hours later, it was only 5 percent completed and still working on two bad sectors it had found. I noticed in the data it was working on, it was Microsoft's, and wondered if the failure had munged a file related to software validation. I decided to just let it run its course.

"About 36 hours later SpinRite had finished testing the drive with a total of about 16 blocks marked bad. When I booted into the system, I was able to launch IE now, and ran Windows Update. After installing all the patches and the software validation tool, lo and behold, the errors were all gone. I got the window message that Microsoft had recertified the box, and all was well with the world again. Thank you, SpinRite, for saving me the trouble of backing up data and reinstalling everything. Hopefully the drive will last long enough now for them to afford a new one. Great product. Mark in Oregon." And thank you, Mark Wright. I really appreciate you sharing that with our listeners.

**Leo:** Time for questions and answers. Are you ready?

**Steve:** You betcha.

**Leo:** Well, of course you are. Why wouldn't you be ready? The question is, is Leo ready? Let me get the document up here, and then I can answer the questions. Where are they? It says they're in there. It seems to end at the SpinRite thing. Is there a second?

**Steve:** Oh, I'm sorry, Leo. There's two documents.

**Leo:** Oh, well, there you go.

**Steve:** They're a separate PDF.

**Leo:** Of course they are. Oh, there it is. All right, I'm ready. Sorry about that, Steve. Sometimes I'm a little slow. Question #1 comes to us from a listener and programmer. In fact, I know his name, Jim Hyslop. He rants against Steve's claim: Steve, in Episode 256, Q&A #115, your very passionate claim that it is possible to have bug-free software is a major slap in the face to those of us who build software for a living. Well, I'm going to let you defend yourself, Steve. But I think that Steve has been very clear about this, and it's not quite as you say.

Sir, you have stained the honor of all professional programmers, and I will have satisfaction. As the injured party, I hereby challenge you to a thumb wrestling duel to be held at a place and time that is mutually convenient. All joking aside, though, you exclaimed, "Come on, it's math!" No, it's not. Good programming is communication - communicating your intentions to the compiler, and more importantly to other programmers in such a way that there can be no confusion. That's why, as you pointed out in your very much justified rant against JavaScript, every browser interpreted a particular code snippet in a different way. Communication is a human activity, and no matter how carefully you choose your words, someone will interpret what you say in a different manner. I can't count the number of times I've had QA testers file bug reports, only to tell them, "No, that's how it's supposed to work. Look at this section in the design document."

And of course people make mistakes, too. Add in the complexities of multithreaded and event-driven programs, you're now talking about programs whose complexity is several orders of magnitude greater than most programs that could be written in assembler. At that point, proving that a program is bug free is almost an impossible task. You said, "By definition it's possible for us to have an absolutely bug-free environment and not a bug in any apps," and I want to underscore this, "but it'll never happen." You said that. Well, it is also possible to win the lottery five times in a row, but that'll never happen.

Just to be clear, I'm not saying because we can't write bug-free code we should just shrug our shoulders and say "Oh, well," or that we should expect such basic mistakes as not sanitizing inputs, allowing buffer overruns, and so on. Good software developers should, nay, must always do their best to write code that is as bug free as they know how to make it. And yes, Steve, I'm afraid that means there will always be bugs, and people will always make mistakes like shutting off a firewall. Well, in your defense, you've always said that. It's exactly what you've always said.

**Steve:** [Laughing] First of all, I also build software for a living.

**Leo:** You know a little bit about this.

**Steve:** Yeah. And he says that "add in the complexities of multithreaded and event-driven programs, and you are now talking about programs whose complexity is several orders of magnitude greater than most programs that could be written in assembler." Well...

**Leo:** Not so.

**Steve:** I write all of my programs in assembler. And they're all multithreaded, and they're all event driven. So I think what he's saying there is that, if you were in assembler, then you're at the bare metal, so you're not depending upon, for example, the whims of the compiler or the JavaScript interpreter and communication. I mean, I certainly agree with him that communication is one of the problems. One of the things I like, for example, about the way the Internet's RFCs, the Requests for Comment, have been structured, is they make a very good point, and they always use all capitals, they say SHALL do this, MUST do this, MAY do this, SHALL NOT, MAY NOT, MUST NOT, you know, they are extremely careful when they're writing these specifications about what behavior they want.

So certainly communication is part of it. Yet I will, at the same time, I will stand by my statement, which is that there isn't anything analog, from the beginning, of the way our computers work. And so, while, yes, it's not going to be the case that massively complex systems written by, when you aggregate all the people involved, tens of thousands of people speaking different languages, thinking different things, meaning different things, and then having it all come together, that it's going to work. In fact, when you phrase it that way, it's amazing these things even boot. Still, from a theoretical standpoint, and this is really what I was saying in Episode 256, is it can be perfect. It absolutely can be perfect. Will it be? No. Can it be? Absolutely. There's nothing preventing our operating systems and our programs from being perfect. Yeah, they're not going to be. But...

**Leo:** Yeah, I mean, in your defense, you've always said that we will never get rid of bugs.

**Steve:** Right.

**Leo:** So there's a difference between making an assertion that theoretically it is possible, as with anything like this, to make it perfect. But practically it's impossible.

**Steve:** Correct.

**Leo:** Would that be a fair way to describe what you believe?

**Steve:** Yes. Practically, it's, well, practically it's never going to happen. It's not impossible, it's never going to happen. I love Donald Knuth's book on, is it LeX?

**Leo:** No, TeX. TeX, it's called, T-e-X.

**Steve:** Yeah, TeX.

**Leo:** He pronounces it "Tech," just to be additionally obscure.

**Steve:** And in the preface he says, "I believe on" - and he quotes a date - "the last bug in this program was found." And I think he says, "And the person who found it got $2.56. And I will double the reward for every successive bug that's found, except I don't think there are anymore."

**Leo:** Wow. That's bold.

**Steve:** And I love that. No, I mean, again, I mean, this is Donald. I mean, this is the guy

- he's an artist of software.

**Leo:** That's the name of his book.

**Steve:** And TeX is massive. I mean, it is a serious piece of code. And it may very well be perfect.

**Leo:** So no one's found another bug?

**Steve:** No. There probably aren't any.

**Leo:** Wow.

**Steve:** I mean, he probably did find them all. Because he wrote it in a language that he knew, well, actually in a language that he invented, and wrote it very carefully, and, I mean, that's just the way he is. Now, is that a commercial practicality? No. I mean, he would have been fired by any employer. But he's the person we all bow to as the master of the art and science of computer programming. Not the economics, not the practical reality. But boy, he knows what he's doing. And he wrote a perfect program.

**Leo:** Okay. One. There's one. There's been one. Pat Cho in Sacramento, California wonders about disabling browser plug-ins: Steve, while I would prefer not to have Java and other - now we agree - Java and other plug-ins installed on my computer, I do need them for a few sites. Firefox and most other browsers give you the option to disable them, which I do unless I need them for a specific web page. Am I gaining any additional security by doing this? Or am I just wasting my time because the malware can somehow access the needed DLL files even if they aren't enabled because they exist on the computer? If this does provide some additional security, I hope someone develops an extension to make it easier to turn the plug-ins off and on. Thanks for the great show, and thanks for SpinRite. I would add it's probably also programmatically possible to tell the browser to turn them back on anyway.

**Steve:** Yeah. I don't know whether you could do that, at least in Firefox. So, okay, Pat's question is a good one, as we just saw, we dragged ourselves over the horrible details of a new rootkit which installs itself thanks to having an obsolete version of Java installed on the machine. And as I mentioned, it is scripting that enables the Java applet to be loaded and run, just as it's scripting that enables Flash applets to load and run.

So it is definitely the case that browsers have control over what technologies, what add-on technologies, plug-in technologies they're hosting and running within their windows; and that it would be possible to do something more granular than NoScript, where, for example, when you load a page, up pops like a menu of the technologies that this page needs in order to go. And there would be a field for Java, and one for Flash, and one for JavaScript, and PDF rendering, and all the different sorts of plug-ins that you have to enhance your browsing experience. And then you could decide which of those, based on this site, which of those you wanted to turn on.

That's sort of what we do with NoScript in a less granular fashion because, without scripting, pretty much nothing has a chance to get going, because scripting is the way everything runs. So it is, though, the case that when you turn scripting on, you are then allowing anything the script wants to do on that domain to do whatever it wants to. I like the fact that NoScript does give you granularity. So you can turn scripting on for that first-party site. But then it shows you a list of all the other third-party contributors to that page and allows you to decide whether you want them on or not. And frankly, I typically leave them off. I look at them, and they're typically third-party advertising or tracking, overtly marketing/tracking-related things. It's like, eh, it'll see if the site works if I only enable first-party scripting. Actually, that'd be sort of a nice feature to have for the browser.

**Leo:** Yeah, no kidding.

**Steve:** Sort of by default. And I know that Chrome does allow you to disable JavaScript and then enable it selectively. It'll give you a warning if the page has scripting, and you can then turn it on for that page, though it does turn on for all other parties to that page, as well. So it gives you some of the feel of what NoScript does over on the Firefox side.

So, yeah, we don't have anything like that. I could foresee a version of NoScript in the future that gives us more sort of easy popup granularity control over these plug-ins because they are now - they're the source of the problems that we have out on the 'Net. It's leveraging mistakes and errors in these plug-ins, whether it's the PDF viewer, the Java renderer, the JavaScript itself. Typically we're seeing less problems with scripting, and it seems to be moving now to the second-level targets which are those plug-ins that the scripting runs. So I think it's a great idea, Pat.

**Leo:** Question #3 comes from Cory in New York City, and his subject is "Police State." Police State. Dear Steve, first of all, thanks for your great work on Security Now!. Now, on to business: I came across an interestingly disturbing article on Ars Technica yesterday. That's actually now a little while ago. Basically, they describe the technology the police can and often do use to grab data from cell phones when they pull someone over. They can do it with a physical connection, or even Bluetooth. I was wondering what, if anything, could be done about this. Would encrypting your phone help? Certainly a TrueCrypt-style encryption would mean nothing useful could be gotten. But are such things readily available or computationally feasible on phones? What about older phones or early generation smartphones like the 3G or the first Droid or the G1? Surely they would take the biggest computational hit for encryption. Is there anything else that could be done? I'm sick and tired of governments assuming that wanting privacy means we're hiding something. Can't wait to hear your thoughts, and thanks for all your work. And he includes a picture on here of a cell...

**Steve:** Actually I did a little research. You see that thing sitting there, this little handheld deal with a little cord going over to the phone. And in the second picture, over on the right, is a bank of all the different connectors which are available for that thing. So get a load of this. While I'm reading this, Leo, Google "Cellebrite UFED." Again, that's C-e-l-l-e-b-r-i-t-e space UFED. So an article in TheNewspaper.com for Michigan says "Police Search Cell Phones During Traffic Stops: ACLU seeks information on Michigan program that allows cops to download information from smartphones belonging to stopped motorists. The Michigan State Police have a high-tech mobile forensics device that can be used to extract information from cell phones belonging to motorists stopped for minor traffic violations. The American Civil Liberties Union (ACLU) of Michigan last Wednesday demanded that state officials stop stonewalling freedom of information requests for information on the program.

ACLU learned that the police had acquired the cell phone scanning devices and in August 2008 filed an official request for records on the program" - that is, the ACLU filed a request - "including logs of how the devices were used. The state police responded by saying they would provide the information only in return for a payment of $544,680."

**Leo:** Why?

**Steve:** "The ACLU found the charge outrageous." Yeah, no kidding. Okay. So I thought, what is going on with this thing? So this device is called the Cellebrite UFED. I tracked it down, went to their site, and I'm looking at a picture with this array of cords and the Cellebrite UFED system real-time mobile forensics. This is a handheld thing. Says: "The Cellebrite UFED forensic system is the ultimate standalone mobile forensic device, ready for use out in the field or in the lab. The UFED system extracts vital information from 95 percent of all cellular phones on the market today, including smartphones and PDA devices - Palm OS, Microsoft, Blackberry, Symbian, iPhone, and Google Android.

"Simple to use, even in the field, with no PC required, the UFED can easily store hundreds of phone books and content items onto an SD card or USB flash drive. Cellebrite UFED supports all known cellular device interfaces, including serial, USB, infrared, and Bluetooth. Extractions can then be brought back to the forensic lab for review and verification using the reporting/analysis tool. Cellebrite" - get this - "works exclusively with most major carriers worldwide, including Verizon Wireless, AT&T, Sprint/Nextel, T-Mobile, Rogers Wireless - Canada, Orange France, and Telstra Australia, as well as 140 others." Get this. "This ensures that future devices are supported prior to retail launch." And then under "Secure Extraction and Complete Content," they say "The UFED allows you to extract a wide variety of data types, including contacts, SMS text messages, deleted text messages, call history, received/dialed/missed, audio, video, pictures and images, ringtones, and phone details including the ESN and the phone number and so forth."

So essentially what we've got is a device designed to hook up to phones, which apparently by design gets in underneath any password protection or encryption that the phones offer because this has been set up in advance with the cell phone providers to make sure this is going to be compatible with every connector shape known so that anyone who gets your cell phone is able to essentially suck it dry. And the problem is, this is being done by just simple traffic stops in Michigan.

**Leo:** Well, that's the issue. I mean, I don't think they have probable cause to search your cell phone for a traffic stop.

**Steve:** Exactly.

**Leo:** But this is similar to a device that they'll use at the phone store to copy your phone numbers off your phone and put it on a new phone.

**Steve:** With your permission. With your permission.

**Leo:** Yeah, with your permission, of course.

**Steve:** As I understand it, the police demand the cell phone of someone who they've pulled over, and then take it back to their car.

**Leo:** Problem is they can do it so quickly, you might not even know it's happened.

**Steve:** Right.

**Leo:** Let's see a court case based on that evidence, then we'll see what happens.

**Steve:** Yeah, we'll see how it develops.

**Leo:** I mean, it's clearly an illegal search and seizure. But anyway, I'm no attorney. I mean, you've got to have probable cause. You can't just take somebody's cell phone, search it on a fishing expedition. That's clearly illegal. That's why the ACLU got involved.

**Steve:** Well, the problem is we are certainly seeing an erosion of our civil liberties as a consequence of the Patriot Act, that was just recently renewed for another four years without any debate in Congress.

**Leo:** Martin Rojas, Atlanta, Georgia wants to set up a secure email community: Steve, I love the show, and I've been listening since Episode 30. I have to say it's made me appreciate what I was learning at my computer science classes and how it applied to the real world. I love hearing your explanations and propeller hat episodes. But recently my friends and myself have been trying to figure out how to encrypt email while communicating within our group. I immediately thought of public key encryption, but I have no idea of any software or how I would go about setting this up for our group. I know most time topics are theoretical, but I think a lot of people would love a practical way to apply encryption to our mail. Love the show. Please keep up the awesome job you and Leo do with the podcast.

**Steve:** And Leo, I turn this one over to you.

**Leo:** I guess because I do it.

**Steve:** Because you do it, and I never have.

**Leo:** Yeah. There's a very easy way. There's a service called Hushmail that Phil Zimmermann of PGP fame worked with. They use PGP's technology in the background. They allow you to create encrypted mail to and from friends or anybody else. It's just like Gmail or any other service, but encryption is one of the features. Now, that may make you a little nervous because you'd have to trust that they weren't in fact storing the key and all that. So many of us just put encryption on our own system. Most email programs allow you to do this. Outlook does. Apple's Mail does. Thunderbird does.

There are kind of two ways to do it. One is with PGP, Pretty Good Privacy. So if you search for PGP, or the one I use which is GNU Privacy Guard, or GNU GP, you'll find implementations for many email programs. That's one way to do it. The other way - and by the way, both these systems will provide you with digital signing as well as encryption. Which means you don't necessarily have to encrypt the mail, but it will validate that the mail came from you and no one else. And that's what I use. If you get email from me, it's always signed.

The other way to sign and/or encrypt is with certificates. And you can go to - there are a lot of places you can buy certificates, from all the usual authorities. Email certificates are usually cheap. I got a free one recently, it used to be Thawte would do this, and I can't remember who does the free email certificates. But if you search around, you can find those. Those are a little bit easier because you install the certificate into your email program, and it handles all the details kind of transparently.

I don't know if Gmail or Yahoo Mail or those other webmails have their own encryption systems. They might. But I would suggest Hushmail if you want to do it with webmail. And I would suggest either GNU Privacy Guard or a certificate-based system, S/MIME is what it's called, Secure MIME, if you want to do it on email. Did I get that right, Steve?

**Steve:** You did.

**Leo:** And I do it, and I like it. Mostly because I think, if we all encrypt, then that just kind of takes away that thing, well, only crooks encrypt; right? The implication is, if you didn't have anything to hide, you wouldn't hide it.

**Steve:** Good point.

**Leo:** So I hide everything. Apparently there are browser plug-ins that will allow you to use PGP with web-based mail. So that's worth looking into. Oh, I guess it's back to me. Question #5 from Tim Roesslein in Saint Louis, Missouri. He wonders about optimum password brute-force strategies. Steve, I'm listening to Security Now! thanks to a friend of mine, Andy Gibson. Good job, Andy. I started from the beginning, and I'm almost up to Episode 100, and I'm excited to find out what the surprise was you promised for that milestone. I don't even remember anymore. That was 200 episodes ago.

Every once in a while I'll sprinkle in a more recent episode, just finished listening to

297. I'm writing this shortly after hearing you say that in terms of password or passphrase vulnerability, the attacker has no knowledge of your character scheme, with Leo adding it might even be foolish for the attacker to make assumptions about it. We talked about tricks you might use that make it easier for you to remember, theoretically would make it easier to crack, if the attacker knew the trick you were using.

He does say: But they have to start somewhere, and that got me wondering if brute-force attacks were tiered. In other words, does a typical brute-force attack in fact start with the assumption of a simple password, perhaps with a limited character set, all lower-case alpha, and then tier up to include upper case, then numerical, ultimately special characters? The bottom line is, if so, wouldn't you be most secure by only picking from the special character set, as that would be the tail end of any brute-force attempts, thereby making the attacker's job more difficult by simply choosing exclusively from the last upper tier.

P.S.: You can't very well listen to 100 hundred episodes of Security Now! without eventually buying a copy of SpinRite, so one of those Yabba Dabba Doos was me. No problem with any of my drives, but it's nice to have a bit-level confirmation they're still in good shape. Grateful for your and Leo's contribution to the field, Jim Roesslein, Saint Louis. We're grateful you listen, Tim.

**Steve:** Yeah, and it's a great question. If I were trying to design a password brute-forcing technology, and we can assume that other attackers understand this problem domain as well as I do, that's exactly how I would tackle the problem. The first thing you would do is use the readily available dictionaries of most common passwords, the things that people most often do, for example, abc123 is, like, right up at the top of the list. And then after doing that you'd run through actual dictionary, like the dictionary in the language of the person whose password you're trying to crack, and see if they just use a dictionary word. And maybe you'd capitalize the first letter, or maybe not, but try it both ways. But, for example, you wouldn't try all caps because that's harder to type in, sort of less likely that that's what they did.

So again, as Tim suggests, you would sort of ramp up your attack, trying successively less likely things, not overlapping the later tests with the earlier ones, that is, skipping those because you've already - you would have tried them earlier. But eventually you'd get all the way out to the kind of password that we've talked about often, which just looks like gibberish. But most people to this day aren't using passwords like that. They're using something much simpler, which is - unfortunately they're using things that are too simple.

But it certainly is the case, exactly as Tim suggests, that the wise attacker would not just start a-b-c-d-e and then aa-ab-ac-ad-ae and so forth, and go through all of them. Instead they'd strategize their attack to maximize the chance of hitting on the solution in the minimum time. That's how I would imagine those attacks would be designed. And I have seen literature that indicates that's what the bad guys are doing.

**Leo:** Well, just as a bad guy can't guess your method, we can't guess or guarantee their method.

**Steve:** Correct.

**Leo:** So, I mean, making assumptions about the method works, but only occasionally. And if you make the wrong assumption, it's going to be worse.

**Steve:** Well, and it's inherent in this that we're talking about a large population of users. That is, the world's users probably are using dictionary-oriented poor passwords, not the Security Now! listeners.

**Leo:** Right. I think why reduce entropy by limiting it to a certain character set. Just throw all the entropy you can at it. Question #6, Levi D. Smith in Oakridge, Tennessee wants his WebGL: This afternoon I listened to Security Now! #300. I was concerned about the comments about WebGL. WebGL is a powerful technology, which provides a standard method for rendering 3D applications in web browsers. I agree there are security flaws in the initial implementations of the standard, but to blackball the entire WebGL API as a security risk is unfair. The focus should be placed on fixing the security vulnerabilities of the browser implementations, instead of rejecting the WebGL library in its entirety. This is actually a debate Steve and I have because basically Steve doesn't like a web page to have any programmability at all. Right?

**Steve:** [Laughing] That's right, it's dangerous.

**Leo:** It's dangerous inherently. So plaintext is always best from Steve's point of view. But we live in a world where we're trying to do more and more with our web pages.

**Steve:** Right. So my answer to Levi is, yes, by all means, we want the implementations of WebGL to get better and stronger with time. My job here with the podcast is to keep our listeners aware of the threats that are lurking. And from a standard security standpoint, it is always the case to operate with the minimum attack surface, that is, the fewest number of things that can be attacked. So I don't think I've ever gone to a WebGL-based site. There are demo sites and demo pages. It's like, whoo, look at that. That's in my browser. It's like, okay. But the sites I go to don't use WebGL.

So today, while it's a known attack vector, while we're working on solidifying it and shoring it up, I'm disabling it. I'm turning it off. And if I go to a site that'll say, oh, you apparently don't have a WebGL-enabled browser, it's like, oh. That's my clue, if I care, to turn WebGL on selectively for that site. It's like a firewall. Firewalls deliberately restrict the incoming ports. They don't have - we know that there are 65,535 possible ports. But the whole reason we have a firewall is to hugely constrict the flow of communications to only the things that we know we want. And so disabling WebGL is like using a firewall. It's like saying, I don't know that I need this. Or it's like disabling Java or JavaScript. I don't know that I need this, and now I know it's a potential vulnerability. It's just crazy not to disable it until it stops being a vulnerability, or maybe ever, and turning it on more if WebGL catches on. Maybe it'll always be just sort of a curio and never be a major factor in the industry, in which case I'll probably just leave it off.

**Leo:** All right.

**Steve:** So there.

**Leo:** So there. I have a feeling, because it's HTML5, we're looking at this as a standard. But we'll see. I think it's part of HTML5. Maybe not. I might be wrong on that. Question #7, an anonymous listener writes: Hi, Steve. I just finished listening to #299, went straight to your page, the JavaScript demo page, GRC.com/r&d/js.htm. Of course I used NoScript, duh. So, best "No JavaScript" warning ever. Oh, I haven't seen it. What did you do? What did you do?

**Steve:** [Laughing]

**Leo:** I should turn off JavaScript and go there.

**Steve:** I got a kick out of his mentioning that because I had forgotten that I had fun with the warning that the page will give you, or the explanation, if you go to either of my JavaScript pages with JavaScript disabled. Which of course I would expect our listeners to do, much as this anonymous listener did. And so I just give people a fun message. So I thought I would share that with our listeners who may be curious now. It's GRC.com/r&d/js.htm.

**Leo:** I'm trying to figure out how to turn off JavaScript without NoScript in Chrome. Oh, well. I'll just have to leave that as an exercise for the viewer.

**Steve:** Yeah.

**Leo:** Question #8, Aaron in Bend, Oregon wonders about USB prophylactics. That's something new. Steve and Leo, I'm sitting here with my thumb drive stuck deeply inside a friend's infected PC, trying various tools to clean it, including the new MS Safety Scanner you mentioned a couple of episodes ago. When I am done and want to use this thumb drive again, what is the safest way to use it in my own computer again, after being in an infested PC? Is it enough to have autorun turned off on my PC? Then I'd format and copy the programs I use back on it.

I did some Googling tonight and found a couple of free programs that claim to make your USB flash read-only. Oh, that's interesting. I also see you can buy flash drives with write-protect switch, like an old floppy disk. But I don't want to buy another flash drive when I have so many lying around, and I didn't find any software from a source I recognized and trusted. I also thought of formatting while on the infected PC after I'm done, but I don't trust malware not to hop back on after it's formatted and before I can yank it out. As always, thank you for the podcast. Aaron. Boy, that's a good point. If you stick something into an infected PC, it might get infected before you could take it out.

**Steve:** It is a fantastic point, and I loved the question. And it raises - it is a great question. We know that malware jumps onto thumb drives. That's how Stuxnet got itself all over the place, and it's a common thing to do because thumb drives are sort of today's version of the floppy that was how the original viruses spread around among PCs, back in the old DOS days. And thinking about this, the only thing I could suggest that is safe - and he's absolutely right. I mean, I would have exactly his reaction. If I was using a thumb drive in a known infected machine, that thumb drive is absolutely suspect from now on. I mean, I'd be tempted just never to use it again, just drop it off in the next garbage can, because it's scary.

**Leo:** Wow.

**Steve:** Well, and because to really fix it you have to jump through some hoops. I would say boot from one of the boot CDs, like an Ubuntu Linux distro CD, instead of booting from your main regular bootable system. He asks is it enough to have autorun turned off, and we know it's not because unfortunately there are bugs in the display of the contents which allow malware to gain control just by, like, viewing the contents in Explorer is all it takes. And that's one of the things that Stuxnet used. And there are still some unknown exploits that have never been made public that were being leveraged by Stuxnet. So we still don't know what those are or if they've been fixed.

So I just don't think it's safe. Unfortunately, we've got too much automation in our systems to allow a USB drive to be plugged in safely. So I would say boot from a boot CD without your hard drives connected, that is, so that there is nothing writeable on this machine except the thumb drive. And that's why I'm saying it's sort of impractical. But frankly that's what I would do. I would disconnect my hard drives, boot a bootable CD, and then use that to reformat the - and don't just erase the files because you could have hidden files. Do a full format of that drive in order to clean it. Or what's easier, I mean, they're so inexpensive these days, I'd just maybe consider this one your malware fixer thumb drive, put it in a red box with the skull and crossbones on it, and use it the next time there's a problem.

**Leo:** No, because then you infect the next machine.

**Steve:** Yeah, that's a problem, too. It really is a good question.

**Leo:** Anybody who does this does not use a thumb drive, they use a CD or a DVD with all the tools on it. I presume the reason he's bringing a thumb drive is because he's got his tools on it. Just burn a CD.

**Steve:** Very good point. Simply burn it to a CD. It's a much better solution.

**Leo:** Admittedly, a DVD's only 4.7 gigs. But I think that's enough.

**Steve:** Oh, I would imagine, yeah.

**Leo:** How many tools do you have? If you need two, burn two, or three. Now, if it's a Netbook, and there's no drive, yeah, I guess you - I mean, it's conceivable there's reasons. But, boy, that's a good point, I tell you. You can't trust it now.

**Steve:** Yeah. Once it's been in a bad drive, I'm not so happy with it.

**Leo:** Frank Varela in Boyle Heights, California wants more on - we're going to call it PIE.

**Steve:** Yep.

**Leo:** It was PEE, Pre-Egression Encryption. Now it's Pre-Internet Encryption. Long-time listener, always fascinated with the topics. You brought up the term PEE - PIE. Could you talk a little more about what is Pre-Internet Egression?

**Steve:** Well, I'm not going to spend much time on this because we have already talked about it. I pulled all these questions together before I put the top of the show stuff together, which brought me into this discussion. So I'm sure all of our listeners understand, and I'll just make sure that Frank does, that this concept is using technology and, for example, Jungle Disk is an example of PIE, Pre-Internet Encryption, where although it's not the default, you have to manually establish your own encryption key, for example, in Jungle Disk. And there are some others that people have been tweeting me about that I'm going to try to make time to look at in order to vet them. Because I'm not really happy with the direction Jungle Disk has taken. It was once free; now it's not. It's still inexpensive for in-the-cloud backup. And I think if you use their Rackspace version, and you use your own key, then you have a PIE-compatible, a Pre-Internet Encryption system.

But the idea is many systems, like Dropbox, are very user-friendly, and they say, oh, we encrypt. We use SSL 256 encryption so that all of your data is safe as it's coming to us. The problem is, they encrypt it, and then they decrypt it at the other end. So they're storing it, or they have it, at least, in an unencrypted state. In the case of Dropbox, they then would encrypt it for storage. But they encrypted it for storage. They have the key that was used. The only way any of this stuff is safe is if you do the encryption before it goes out on the wire, and that key never leaves your control. In which case we're using the cloud as a big opaque storage container in the sky.

We still have the problem that it could go offline, and so that's inconvenient. But at least we have zeroed the problem of security. This stuff is absolutely secure because it was encrypted before it left us. And encryption these days is trivial. It's available. It's inex- well, it's free, not even inexpensive. So it does require that you carefully choose and vet the technology you're using. I ought to mention, this is why I did the same thing with LastPass. I'm using LastPass. I understand how it works. And it is PIE-compatible, Pre-Internet Encryption. It encrypts everything that we entrust to LastPass. They never get the key. Which is why it qualifies. Dropbox didn't, and they got caught because people now understand that Dropbox employees, or Dropbox when served with a subpoena, could decrypt the data that is in our Dropbox. And so that's not PIE-compatible.

There are systems that are. And I believe in the future, as we become more security aware, it'll be something that's made more clear. It's very frustrating when I look at something that's got a beautiful-looking website, oh, clearly they spent a lot of money on the graphics, but they don't tell me anything about how the thing works. In which case I can't say, oh, this looks great. It's like, yeah, look at the pretty buttons, wow.

**Leo:** Most people judge the quality of software by the quality of the UI, and that's why, don't you think?

**Steve:** Yeah.

**Leo:** Our last question, #10, comes from Kevin Yong in Los Angeles. He asks about password strength and dictionary attacks: I'm a fan of Security Now!, had a question about password strength and dictionary attacks. I know from your past advice that any normal word used as a password can easily be cracked in a dictionary attack. Does the same hold true for a dictionary word with alphanumeric additions mixed in, such as "eXample05%"?

Also, what about longer passwords containing a mix of dictionary words with numbers and symbols? For example, if my 20-plus character password was something like "I Can't Remember," but replaces one space with a tilde, another with an asterisk, and uses a back tick instead of an apostrophe, and an exclamation at the end, and then adds brackets and #8, would it still be vulnerable to a dictionary attack or a similar brute-force hacking? I guess what he's saying is I can see a dictionary word in there, but it's mixed up, it's muddled up with other stuff.

I'm trying to strike a balance between password strength and memorability and being able to include words or phrases within the mix of alphanumeric characters would make things easier for me. I don't want to make it easier for hackers, too, though, especially if I use it for something like a LastPass master password. Thanks for any advice you and Leo might have.

**Steve:** Now, Leo, you would think that we had beaten this thing to death.

**Leo:** It's a good question.

**Steve:** It is a good question.

**Leo:** Which we've answered.

**Steve:** And I stunned myself Sunday.

**Leo:** Uh-oh, what happened?

**Steve:** With a breakthrough in password technology.

**Leo:** A breakthrough in password technology?

**Steve:** I know how loony that sounds. Again, you'd think we had, I mean, in the 302 episodes, we would have - I mean, on our first episodes, way back, 1, 2, and 3 were on passwords. And we've mentioned passwords over and over and over because, like it or not, they're the way we all authenticate, still, to this day, the majority of us, on the Internet. I mean, they have...

**Leo:** It's probably the most important security thing we deal with day in, day out.

**Steve:** Yes, it is. And so earlier this year I came up with an idea, and I mentioned a couple weeks ago the Passcode Designer that I was working on. It's why I taught myself JavaScript. And you can see it, Leo, if you go to GRC.com/passcodedesigner, all run together, that's passcode, not password, so passcodedesigner. You can tack on a .htm if you want. If you don't, my server will. And it's a little machine that I built over the last month or so, very cute and graphical. You can click on things. You can type in the field. You can play with it.

**Leo:** And it's got a big "obsolete" stamp on the front of it.

**Steve:** Yes. Not only is it obsolete, but GRC's Perfect Passwords are obsolete.

**Leo:** What?

**Steve:** Everything is obsolete. I'm not kidding you. This is unbelievable what I came up with. And I've been just, like, reeling from it for a couple days. Now, when I use my own passwords, I think, boy, is this stupid. I've got to get this changed to the new scheme.

**Leo:** You have a new password scheme.

**Steve:** A whole new, I mean, this is unbelievable. Next week we're going to change the whole balance between strength and memorability in a way that means we can now type in - we'll be able to enter our WiFi router passwords. They will be as strong as if you used a Perfect Password from GRC. But it's vastly simpler.

**Leo:** Oh, thank you. So this is just something you were noodling around, and it came to you in a flash?

**Steve:** Well, what happened was, and this is the nature of research, is I built this machine, this Passcode Designer. And the concept behind it, I thought, okay, well, we know that, like, if you used all lower case, that's obviously weak. And so what that says is that having other classes of symbols, like numbers of special characters, is important. And so I thought, hey, how about if we treated it like a state machine, where we have four different classes. We have lowercase alpha, uppercase alpha, we have symbols, and we have numbers. And you want to encourage the user to, in their password, have transitions between those classes. So on that concept I built this passcode designer, with the goal being to create maximum entropy, minimum length, maximum strength passcodes.

And when I got all done - and it's finished, it's there working. I realized when I got finished it was wrong. And I got stalled for a couple weeks because I couldn't figure out what was wrong, but my intuition was, like, itching me, saying, okay, this isn't right. And I showed it to the folks in the newsgroup. And a lot of them, you know, we went back and forth and tried to figure out - they didn't quite understand what I meant, and I explained it. And so I ended up figuring out what was confusing me. And then on Sunday I got it. And it's like, oh, my god, this changes everything. So nothing I've ever said about passwords...

**Leo:** I can't wait to listen next week.

**Steve:** Nothing I've ever said about passwords is right. I mean, nothing everyone - anyone thinks. I have got some news. I know it sounds like I've lost my mind. But I think I can - I'm working on a new page now which is going to lay it all out and explain it and give people something to play with so they can test passwords using this new scheme. And when you hear it, you're going to go, oh, my god. Why didn't anyone ever think about this before?

**Leo:** Oh, my god.

**Steve:** So Episode 303, next week.

**Leo:** I can't wait. What a tease. I love it. Next week.

**Steve:** I'm not kidding. We're going to change our passwords. Everyone here who is listening to this is going to change their passwords.

**Leo:** I can't wait.

**Steve:** Because I've got something far, far better.

**Leo:** And this is based, because we've been kind of having this conversation all along, and then it was stimulated by that article about why something is, "joy is fun" or something was a better password than xyzzz. And so we've been doing this thinking lately.

**Steve:** Oh, we've talked about entropy. And, I mean, the Perfect Paper Passwords concept is still a good one, the one-time pad, because that's different, the one-time tokens. But GRC's Perfect Passwords, that 3,500 people a day go to, it's just junk. Don't need that anymore.

**Leo:** I presume you will be replacing it soon as we do the show.

**Steve:** It's sad because I really liked it. But it's stupid.

**Leo:** You put a lot of work into it.

**Steve:** Wait till you - I've got something so much better.

**Leo:** Steve, you're the greatest. Steve Gibson is at GRC.com. That's the place to go if you want to know more about SpinRite and all of those other great applications, many of which he gives away. Actually, SpinRite's the only one he charges for. GRC, the Gibson Research Corporation. If you have a question, we do Q&A episodes every other episode. Go to GRC.com/feedback. You'll find all of the Security Now! episodes there in 16KB versions for the bandwidth impaired, plus Steve pays to get transcriptions made so you can read along, which is great. It also makes it easier to search for the content you're looking for. That's all at GRC.com. Steve's on Twitter, @SGgrc. And every week we do this show, Wednesdays at 11:00 a.m. Pacific, 2:00 p.m. Eastern time, 1800 UTC, at live.twit.tv. So join us live or subscribe at TWiT.tv/sn, and you can get it every week. You wouldn't want to miss one, that's for sure. Not next week, anyway.

**Steve:** Not next week. Not #303.

**Leo:** Somebody in the chatroom, Beatmaster is saying, "Could you sell that idea to Sony, please?" All right. We'll talk again next time on Security Now!. Bye bye, Steve.

**Steve:** Thanks, Leo. We're going to give it away next week.

**Leo:** Yay, it's free.