What a Busy Week!

Description: We have so much security news and information to cover this week that we didn't have time to take questions from our listeners. What we have, instead, is a LOT of interesting news about the new Java vulnerabilities, new TNO cloud storage solutions, and lots more.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-367.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-367-lq.mp3

SHOW TEASE: It's time for Security Now!. The Explainer in Chief, Steve Gibson, is here. He's going to talk about a big Java exploit that affects not only Windows, but Mac and Linux, as well, and give you his take on the Samsung/Apple verdict. He is, after all, a software designer. It's all coming up next on Security Now!.

Leo Laporte: It's time for Security Now! with Steve Gibson, Episode 367, recorded August 29, 2012: What a Busy Week!

It's time for Security Now!, the show that covers your safety and privacy online with this guy right here, our Explainer in Chief, Mr. Steve Gibson of GRC.com, the inventor of a lot of very useful security tools and, of course, the ultimate hard drive maintenance and recovery utility, SpinRite. Hi, Steve.

Steve Gibson: Hey, Leo. Great to be with you this week.

Leo: What a week it has been.

Steve: Well, in fact, we've got so much to talk about, just sort of across the board, that I was looking at - this is nominally a Q&A week. But I thought, okay, there's just no way we have time to take any of our listeners' questions. So I'm going to push that to next week, and this week we'll just talk about everything that's happened and has been going on. I have found - we need to talk about the big Java problems which people need to be aware of. There's some new malware that is able to infect virtual machines images, VMware images at rest. That is, when they're not in use, this will go and infect the static file. So it's like, okay, well, I guess that was foreseeable, but it now exists.

Dropbox made some news with the addition of two-factor authentication. I've been playing with it and have implemented it, and I'll talk about that. And I found some new Trust No One cloud storage solutions. So, and a bunch of all kind of miscellaneous stuff. And I want to talk to you a little bit about the Apple vs. Samsung thing. I don't have a strong position. I'm not screening in either direction. But as a person who's gone through the patent process a number of times, and an innovator and developer, I thought it would be fun to chat with you a little bit on the consequence of all that.

Leo: I would love to hear that.

Steve: Lots of stuff.

Leo: Our expert witness, so to speak. Great. I think we can launch right in, actually. We have but one commercial. We'll do that before, well, we'll just figure out a spot.

Steve: Yeah. Okay, so I got a kick out of one, I guess sort of a hacker code tester person, I can't remember. His tweet handle is @0xabad1dea.

Leo: Ooh.

Steve: As in "a bad idea." Anyway, he coined the term and our friend Simon Zerafa, who watches a lot of feeds, apparently, retweeted it, so I saw it. He called this the "Javapocalypse." What we have is we have a bad new problem in Java, such that everyone is now being advised to remove it.

Leo: I love the, who was it, the guy in charge of security at F-Protect who, what was his quote? It was really quite funny. It's Windows, Mac, and Linux.

Steve: Yes. Well, and this is the mixed blessing of Java. I mean, the whole point of Java is you write it once, and you run it anywhere. And so what happened is that, with the release of v7 - we recently went from 6 to 7, and 6 is still being updated a little bit for people who haven't moved yet to 7. But two new classes were added to 7. And it turns out that there is a very clever way of leveraging some mistakes which were made in 7, which have been around for a while. So no one is sure where this may have been used before, but it suddenly exploded into public awareness over the last couple days because, well, because it's now being used for targeted attacks.

The typical, you browse somewhere with a computer that has Oracle's Java 7 installed. And I'm a little confused about, because I keep seeing - I've seen conflicting reports. Some have said that you have to have the very latest version, v1.7. Some have said that 1.6 is not vulnerable. Other people have said, oh, no, all versions of 7 are vulnerable. We do know that this happened with the move from 6 to 7, so 6 is not vulnerable to this, but it's vulnerable to other things. So no one is recommending people go to an earlier version of Java. I mean, it's a mess. So it's, as you said, all three operating systems - Windows, Mac, and Linux. Now, it's interesting, the Ubuntu doesn't come normally, for example, with Oracle's Java. They've got their own OpenJRE, the Java...

Leo: Just like Apple. Just like Apple.

Steve: And it's not vulnerable. So if you use the Java that's natively in Ubuntu, for example, you're okay. And one of the people exploring this had to remove that, then install the latest version of Oracle's Java, and then was able to make the exploit happen. There were some early reports that Chrome was not vulnerable; but, I mean, this has been moving very fast. I mean, just minute by minute, hour by hour. It's already in the Metasploit framework. It's already in the Blackhole rootkit, which is used by bunches of bad guys. So it's completely available. There's a full technical explanation that shows the source code, step by step of how this works.

There are some security experts who are unhappy with what they consider to be irresponsible disclosure of this. I mean, everyone just kind of went crazy. Some people early on were actually posting links to infected websites which could have infected people who clicked on those links. So there's been a bit of a frenzy about this because it's any browser, any OS. Now, currently it's only carrying Windows malware. But the way this works is interesting, too. Leveraging these two particular methods in two classes, what this essentially allows is the applet to modify its own operating system security settings. So it's not a buffer overflow or anything like that.

There's a way for the applet that you would download a .JAR file when you clicked on a link, went to a website or opened email or something. Normally there's containment of the Java environment within your browser so that it can't arbitrarily do things to your system. So the applet doesn't have read, write, and execute permissions on the operating system file system itself. But what these two methods allow, when used together, is the applet to give itself unrestricted permission to read and write and execute code. So when it has that, it then goes out and gets another executable, I think in this case it's called "hi.exe," which it grabs. And at this point it's a Windows-only exploit. So although the vulnerability exists on the other platforms, it's not currently, in the wild, as far as people know who have seen it, going against Mac and Linux machines. But it's fully capable of doing so.

In the Windows case, it overwrites a file in the Windows/system32 directory where all the components of Windows are kept. And this is the portable media serial number service that gets overwritten. It's an awkwardly named file, mspmsnsv.dll, that gets replaced. And at the moment what it does is it downloads and installs the Poison Ivy RAT. RAT is an acronym for Remote Access Trojan. So that's what people are being afflicted by who click on links in their browsers who have Java enabled and running.

So once again, this is essentially a call to disable Java. Now, as we know, Apple has already made the move, as a consequence of the catastrophe they recently suffered where so many hundreds of thousands of Mac machines were infected because of Java, where it's disabling it by itself. You have to manually reenable Java. And then if you don't use it for a while, it goes to sleep again and redisables itself. Unfortunately, it's really looking like this is the only way we're going to be able to coexist with Java. It's a shame because Java is so powerful. It can do many things. There are organizations that are critically dependent upon Java. They've written big chunks of their own stuff in Java because they were convinced that it was a good thing.

There is available a patch. Now, the problem is that Oracle has a four-month patch cycle. They only patch, not even quarterly, but three times a year, every four months. And so that was the middle of February, the middle of June, and the middle of October. So here we are approaching the end of August. We've got all of September and half of October.

It's October 16th is the official next patch cycle for Java. Everybody has been asking Oracle, what's going on, are we going to get an out-of-cycle patch, are they going to fix this quick. And so far, silence from Oracle. No one that I'm aware of who has asked has received any answer.

So this again, we've talked many times about opening up Add/Remove Programs, looking at plugins in your browser. If you don't know you need Java, it's difficult to justify it in this situation. We don't know how soon it's going to get fixed. There is a website which basically just checks your version number, but it's IsJavaExploitable.com. So you can safely go to IsJavaExploitable.com. Now, I went, and because I've got NoScript running in Firefox, nothing happened.

Leo: So this is from Metasploit, so they're a reliable company.

Steve: Yeah. And if you enable scripting, then it will use Java to report on its own version number and tell you where you are vulnerable, not by executing the vulnerability, but by just looking at the version number because, if you've got 1.7, and you've been up to date, and unfortunately that's what I had when I enabled it, then it's like, oh, yes, you are vulnerable. So our friend Brian Krebs at KrebsOnSecurity.com recently blogged how to unplug Java from the browser. So if anyone who listens to the podcast is not already clear about going to Add/Remove Programs in Windows and removing it and then looking at your plugin, your browser plugins, and either disabling it or removing it, Brian does have some nice browser-specific step-by-step instructions on a recent blog of his. So it's KrebsOnSecurity.com.

Leo: So if you're using Ubuntu or using Macintosh, you're probably all right; right? Because you're not using...

Steve: Well, that's a good question. What happens, for example, on your Mac? I didn't try it yet on mine.

Leo: Mine, I just ran it, and I have an up-to-date Mountain Lion installation, and it said it's not exploitable. But I always get from Safari the warning before Java runs anyway.

Steve: Yes.

Leo: Which is helpful because, if Java suddenly - if you go to a website, and it says I want to run Java, you might want to say no.

Steve: Yes. Now would be a good time to say no.

Leo: Yeah.

Steve: Now, I've listened to quotes from hackers, read quotes from hackers, and they're

just jumping up and down because they feel like they have a six-week window, minimum.

Leo: Woohoo!

Steve: Yeah, I mean, it's not often that you get a completely cross-platform, wide-open exploit which is able - and what this will do, let me just make it clear, it will not just download a Windows exploit. I mean, all it is, you could easily have a multiplatform exploit. It will download anything and run anything on any system. It has that capability. Essentially, it completely unchains a programming language, Java, from any security constraints, allowing it to do, like an app running on your machine, with you doing nothing but visiting a website, if you have no other protections in place. So you can imagine the script kiddies and the hackers, it's like, oh, boy, here's an opportunity for some creativity. Ha ha, how are we going to be able use this?

And again, within the range of this podcast, people are probably going to be safe. The problem is that not everybody listens to us, Leo, and there's going to be lots of people probably caught out by this. And again, Oracle's not saying when they're going to have any fix for this. The good news is, it's not installed, for example, in Windows. Normally you get it because you have done something in the past that required it, and so you installed it. Windows doesn't come with Java.

Leo: Same with Macintosh. Doesn't come with Mac either, yeah.

Steve: Yeah. So that certainly lowers the attack surface, and that's good. Although for corporations, which are known to be reliant on Java, this is probably going to get applied in strategically targeted attacks. Now, I did want to mention that a patch has been independently developed at DeependResearch.org. They've got a patch. And if you are not an end-user, they're not going to help you. But if you're a major corporation that is dependent upon Java for your operations, so that it's just not feasible not to use it until Oracle fixes it, you can send email to admin@deependresearch.org and explain who you are, what your need is, and they will provide you with a link to a patch. Essentially it's changing the problems with these two now well-understood problems. So it's trivial to fix. And so everyone's feeling is this is something Oracle could jump on immediately and get to.

Now, the other problem is that, when I recommended that people have Java look for updates more quickly, I have noticed, and I've had other people confirm, that when Java updates itself, it resets its "how often should I check for updates" to - I'm thinking it's a month. So the problem is that, even when it's been fixed, Java isn't going to be looking more often than a month. So...

Leo: That's really not often enough. That's just silly.

Steve: Yeah. It's not. Now, we are also seeing, and maybe the browsers will step up, we're seeing, as we know, browsers being more proactive about checking for the vulnerabilities based on versions of their own plugins. So you could, for example, easily see Mozilla step up to the plate immediately with their evolving on-the-fly patching, just as Chrome does, and preemptively warn people that they've got a vulnerable version of

Java, please fix it as soon as there is an update. So I think we're going to end up talking about this over the next few weeks. This is looking like, unfortunately, because it's so pervasive and multiplatform and such a powerful exploit, I mean, there's nothing more that a bad guy wants than to be able to run any program of their choosing on any machine that visits, that clicks a link in email or visits a website.

Leo: Someone in the chatroom said, "Write once, exploit everywhere." True.

Steve: Yeah. And I don't know what the future is for Java. I mean, we're seeing JavaScript really take hold. It's been standardized. It's becoming increasingly powerful. It's not clear that Java is not going to sort of fall the same way Flash has. It was necessary for a while, but it's not clear that you want something that is powerful enough to do what Java does, that is, web exploitable. I mean, the beauty of JavaScript from a security standpoint is that it doesn't have a file I/O or socket-level I/O where it can do packet things. I mean, JavaScript was designed to run in a browser environment and be limited. Java is a full-fledged programming language that is constrained in an applet, in a downloadable applet environment. But it is inherently powerful.

So what's happened is it's broken free from its constraints in this instance. And so we see the danger of having a full-fledged programming language that can be invoked from a browser link. It's just, I don't know how that's ever going to be a safe thing to do. So it seems to me that what we're going to end up evolving towards is, I mean, and Flash has the same problem. Flash is very capable. It can do lots of things. The problem is, it's invokable from a browser. So I think we're going to, in the future, we're going to see JavaScript continue to mature, its speed improve, its capabilities get fleshed out, yet will always be, hopefully, constrained so that it's safe to run in a browser, and then we really want to move away from high-use plugins that are general purpose programming languages. It's just not safe to invoke them through a browser by clicking on a link anywhere.

Leo: Yeah, wow.

Steve: Yeah. So...

Leo: Unfortunately, one of the most popular games in Existence, Minecraft, is a Java game. So that's on a lot - a lot of people have Java for that reason. Citrix apps are also Java.

Steve: And it's not a problem to have it on your machine. It's a problem to invoke it with a browser.

Leo: Connect it to the browser is the mistake.

Steve: That's, yeah, exactly.

Leo: So you can keep Java. Disable the browser connection.

Steve: Yeah, I mean, you can also run "C" programs on your computer and VBScript and all these other things. So it's just that it is so typically, I mean, when you install Java, it installs its browser plugin.

Leo: Right. That's not a good thing.

Steve: Because it wants to offer those services. I mean, unfortunately, it's funny, I saw someone say three billion machines are vulnerable.

Leo: Yes. Oracle said that.

Steve: Well, because that's how many copies of Java are loose at the moment, unfortunately.

Leo: So disable - in fact, you can do this in most browsers. Just uncheck the Enable Java box; right?

Steve: Yes.

Leo: And there's no reason, you probably don't want it in your browser unless you're, I don't know, playing Yahoo! Games or something.

Steve: I saw a posting that Larry Seltzer had posted in June of 2010, so a little over two years ago. You and I both know him, a great longstanding columnist for PC Mag. And he experimented with removing Java, and his comment at the time was that - and this was two years ago - that, eh, nothing really much happened. Nothing broke. And he said The Wall Street Journal was using Java to display some of their financial charts, but other than that, eh.

Leo: That's because he's not a gamer. A lot of games, RuneScape runs in the browser and is an MMO that runs in the browser. But most modern browsers, including Chrome and Safari now, will not run Java automatically, but will say, this program is asking me to run Java, should I? So just say no unless you're running RuneScape; right?

Steve: Well, and, okay, but here's the problem. We know to say no.

Leo: Right, right.

Steve: But, I mean, it's so funny, I love Jenny, but whenever we're - when, like, we're working on her laptop, things pop up, and she reaches to just click and make it go away. Just, oh, it's in the way. I go, "Wait, wait, wait, I read these things." And she says, oh, okay. But, I mean, I don't mean to pick on her, obviously. She's typical. I mean, this is what everyone does is...

Leo: Yeah, let's get rid of Java.

Steve: Yeah, we really have to.

Leo: Now, here's the interesting thing. Java, when it came out, was really billed as secure, as sandboxed and everything. And I don't understand what went wrong.

Steve: What went wrong was that it's very much like the firewall model. You have an inherent problem when you have capability that you want to restrict. It's better not to have the capability. And so the problem is we have a general purpose, powerful, state-of-the-art programming language, Java, which can do anything. You can write powerful applications in Java. I mean, full-on, standalone, multisystem applications. However, it's the browser component, the idea that, oh, look, we can also use it for web apps.

Well, the second you do that, the second you allow the browser to have access to a full-strength programming language, you're asking for trouble. So they solve that by saying, oh, no, no, don't worry, we're going to restrain its security. We're going to take away, for example, its ability to read and write and execute programs on the hosting system's file system. And everyone says, oh, well, that's good. Well, except that what just happened here was a way around that, where the applet was able to give itself permission. Which is really not what you want.

Leo: Right, right, right.

Steve: So I drew the example of a firewall because, of course, if you have insecure services behind a firewall, I mean, then - and this was the history of Windows in the early days, before there was even a firewall, it was just one problem after another because people would keep finding ways to execute their own code through buffer overruns on Microsoft servers because Microsoft had all these services running on consumer machines that had no need for all these services. They were all just exposed to the Internet. Then we got the firewall, but it wasn't turned on by default, so we might as well not have it. And it wasn't until SP2 of XP that it was on by default. And overnight the problem was gone. Still, a little worrisome to have vulnerable services behind the firewall. But at least we have one.

So in this case I think the lesson we're learning painfully slowly, over and over, is we cannot give our browser access to a full-strength, full-feature programming environment like Flash and like Java. JavaScript was designed for the browser. And arguably Flash was, although it also runs separately. Those are just too powerful. There's just no good way to do it.

Leo: It was the problem ActiveX had. It's just you don't want a browser running arbitrary code on your computer.

Steve: ActiveX the same problem, exactly. It was like downloading a DLL and saying, okay, here you go.

Leo: And we should once again say that the issue is with Java browser-based applets as opposed to standalone Java applications. I mean, the same thing could happen with a standalone application, but you'd have to download the malicious application directly and intentionally.

Steve: Well, and you could have Java on your system used for application execution, as long as you did not have the browser plugin component present and installed. Now, the problem is that Java tends to aggressively reinstall these things in the same way that it turns back its "how often should I check for updates" back to a month, even if you say, oh, I want you to check nightly. If you come back after a while, you realize, oh, wait it's gone to a month again. So it's not behaving itself very well.

I think when the history books are written, Flash will have been a problem, Java in browsers will have been a problem. It was something we may have needed at the time, probably never very well advised. But as no one knew what the Internet was going to become, how pervasive, and how many people who were not computer experts would be casual users of it. And people do just click on - they just click on yes, yes, okay, fine, I want to - get out of my way, whatever you are.

Leo: [Sighing]

Steve: So I saw this little note, I got a kick out of it, that SANS, the SANS Security Institute, caught a story that I wanted to share, just because I keep saying this. But I saw it, but they had it in a story that ComputerworldUK and Yahoo! Finance covered. They said: "An annual survey of 11,000 public company directors and 2,000 general counsels shows that, for the first time, data security is now a prime concern for U.S. boards." As in corporate boards of directors.

"The survey, conducted by advisory firms Corporate Board Member and FTI Consulting, shows that over half, 55 percent in their survey, of general counsels surveyed rate data security as a major concern, while 48 percent of the directors surveyed felt the same. A similar survey in '08" - so four years ago - "found that only 25 percent of directors and 23 percent of general counsel noted data security as a high area of concern, which reflects a doubling of this concern in four years."

And then the president of Corporate Board Member group said about the results, "'While a number of companies are taking steps to become more educated on IT risks, the fact is that not enough are taking the appropriate actions to fully prepare their organization.' He went on to say, 'I think it is going to take several well-publicized security breaches before a majority of corporate boards finally embrace the fact that doing business today without a prudent crisis plan in place is a formula for disaster.'" And I had to read that last sentence a couple times and say, wait a minute, a crisis plan? How about planning not to have a crisis?

Leo: How about that? Well, you should do both. You should try not to have one, but you should have a response plan; right?

Steve: Yes, absolutely. And in fact, one of the ways people or corporations are getting themselves in trouble these days is when they respond poorly to a problem. How they respond is as important as the unfortunate fact that they're being forced to respond to something at all. I'm just - I'm amazed, Leo, at the inertia, at just how slow this is to move forward. But as I keep saying, with all of these widely publicized breaches, these companies are being embarrassed, and that may be the only way that the IT departments are able to get the money and the staffing that they need. Because when you talk to the IT guys, they're like, yes, we're jumping up and down, we tell them all the time.

Leo: We're begging them, please.

Steve: Please, please, please. But instead it's like, well, okay, we'll talk about that next quarter because right now we have different priorities. So it's like, ah, okay, right.

Leo: I think the decision was made at some point by a lot of banks and so forth that they're just going to take a certain amount of loss. You know?

Steve: Yeah.

Leo: This is just the cost of doing business. We can't really stop it. It's like shoplifting.

Steve: Well, look at the credit card fraud problem.

Leo: Totally, they totally accept a percentage of loss.

Steve: Precisely. They just go, okay, well, a certain percentage of these charges are going to be fraudulent, so we'll just bump up the interest rates - and use that as the excuse, by the way, for bumping up the interest rates.

Leo: Very convenient.

Steve: And cover our losses that way.

Leo: Yeah, exactly.

Steve: So "Crisis" is the name of a new piece of malware for Windows which was

discovered last month, in July. It's been found to be capable of infecting VMware virtual machines as well as Windows mobile devices and removable USBs. When originally discovered, Crisis was thought to target just Windows and Mac OS users. It has the capability to record Skype conversations, capture traffic from instant messaging programs, and track websites visited in Firefox or Safari. Symantec says Crisis "searches for a VMware virtual machine image on the compromised computer and, if it finds an image, it mounts the image, then copies itself into the image by using a VMware Player tool." So there's a new first, folks.

Leo: Wow.

Steve: How many times have we talked about virtual machines being one sort of safe harbor, a means of testing viruses and creating some containment and something that we had some control over. But unfortunately, as they've become increasingly popular, it was probably foreseeable that we would end up with a virus that would mount the image, then infect it, and then dismount it so that...

Leo: It's brilliant.

Steve: So that then you fire up your VMware thinking, okay, now I've got a brand new clean thing. And in fact, when it wasn't even running, behind your back it got infected with this thing. Amazing.

I noted that California legislators on both sides of the aisle overwhelmingly passed the Location Privacy Act of 2012. It's a new bill requiring law enforcement agencies to obtain a warrant before collecting any GPS or location data...

Leo: Yes. Hallelujah.

Steve: ...from cell phones or smart phones. It was co-sponsored by the EFF, our friends at the Electronic Frontier Foundation, and the ACLU, the American Civil Liberties Union. And it's been passed now to California Governor Jerry Brown for his hopeful signing into law. He did refuse to sign something last year that was related. I can't remember what it was now. It came as a disappointment and a bit of a surprise. Hopefully this is constrained enough that he's willing to do it.

The EFF had a statement. They said they "urge Governor Brown to have California take the lead on this issue and sign SB 1434" because it "strikes a sensible balance between keeping the public safe and preserving our privacy." So the ACLU did a study, I'm just pulling this from memory, I think it was 383 agencies across the country, law enforcement agencies, they did a Freedom of Information disclosure act request, and more than 200 were in fact using warrantless tracking, which at this point can be done. Some were applying for warrants; others weren't.

And so apparently, as I remember reading this, the fact that some were was taken to mean that everyone could, if we asked them to. So I know that I saw this story also in the SANS security news, and one of their editors was quoted saying, "Another example of California leading the nation in sensible cybersecurity legislation." I understand that law enforcement has a hard time with all this technology. But our Constitution and freedoms

require that there be a balance.

Leo: Hey, you know, it's - you can get a warrant.

Steve: Yeah.

Leo: Just get a warrant. I know it's a pain. You know, it's a pain. Get a warrant.

Steve: So Dropbox has added second-factor authentication.

Leo: This is good.

Steve: It is good. Dan Wheeler blogged two days ago, on August 27th. He said, "Hi, everyone. A few weeks ago we discussed a number of steps we're taking to add an extra layer of security for Dropbox users. Today we'd like to announce the launch of two-step verification, a feature that will enhance the security of your Dropbox by requiring two levels of authentication: your password and a security code that will either be texted to your mobile phone or generated by a mobile authenticator app, available for iOS, Android, Blackberry, and Windows Phone 7."

Okay. So I jumped on this and played with it. I haven't actually been an active Dropbox user, Leo, since you and I stopped using it when we switched to Pogoplug. We were using Dropbox for a while, and I just...

Leo: That's right. I made you use it. What do we use now?

Steve: Now I just go grab the high-quality version.

Leo: Okay, and you just work from that. Okay.

Steve: And just work from that, so it's easy.

Leo: Sorry. I didn't realize - I had kind of lost track of all that. There are people now, you know.

Steve: Well, actually they are great about getting an edited version to me, like within hours. And then I'm able to - see, the problem is Elaine is out in the boonies somewhere with a satellite connection with bandwidth caps. And so downloading 64MB files chokes her. And so if I can bring it down by a quarter of that, it's worth her while for me to do that, so I'm happy to.

So, okay. It is a good thing. What I'm really thinking we're going to see is I think we're going to see a movement to the TOTP, the Time-based One-Time Password. This is the

Google Authenticator. This is a version of the football that we talked about eons ago when PayPal and eBay adopted it. This is the idea that every 30 seconds the code changes, and it's just - you need to have an accurate clock. You need to be synchronized. But the Internet provides time now, so you can expect that things know, your phone knows what time of day it is. And we know that there are even means for achieving synchronization when there is a mis-sync, so that can work.

Google Authenticator is of course open source and free. This TOTP is on the OAuth spec, and it's public domain. It's RFC'd. It's a well-known algorithm. So all of this is open and secure. The idea is that you have a clock which runs through a crypto. So this is a keyed sequence of six-digit changing values. And so the idea is something you want to authenticate with gives you the key. You give the key to this Authenticator, and it's then able to generate the changing six-digit codes that the site you're wanting to authenticate against expects.

So this is nice because the problem that I had with VeriSign, I loved what VeriSign did with their VIP program, it's that they were the single point of failure. Everybody running through them depended upon them. So that was one problem. If they got DDOS'd or went down or got broken into, then there's that problem. The other is they're very expensive. So you don't see lots of people using them because they charge per authentication. So it's great for them if they can convince people to use them. But, boy, it's difficult to use.

Here what we're seeing, instead of having a single device where you use a single third-party service to authenticate, what Google has done is they're saying we'll do like a multi-account authentication. So you can add as many of these keys to Google Authenticator as you want. And so I've got one now that's for Dropbox. And it shows me what the six-digit code that Dropbox expects me to be able to provide when I want to log in. Now, I also - I haven't messed with Windows Phone 7. Maybe you know this, Leo. They talk about something called "Authenticator." Is it just part of Windows?

Leo: I don't know.

Steve: I don't know, but...

Leo: I don't know. I mean, there's Google Authenticator. Maybe there's a Microsoft authenticator. Maybe the chatroom knows.

Steve: Well, now, Google Authenticator is Android, iPhone, and BlackBerry.

Leo: Right. So there's no Windows Phone version of that. So there must be some - maybe it is a VeriSign or something.

Steve: Well, probably not because it's going to be open, where VeriSign is not.

Leo: Right, right, right.

Steve: But what I'm aiming at here is I bet you we're on the cusp of seeing this time-based one-time password system built into our devices. I mean, it's nearly the optimal solution, where you're able to create a code, and this is a one-time password, device-based. Everybody's got a smart phone, pretty much. So the problem with Dropbox is it's a little bit tricky to get set up. They will put up a QR code that you can scan, and I snapped it with an iPad, and it immediately registered. In fact, while the little scanner was open, as soon as it locked into it and focused it, Google Authenticator said, okay, I got it, and showed me the six-digit code that Dropbox then was asking for to confirm that I had received it.

The problem is you can't - I was unable to, and I poked at it for a while, to ever get it to show that to me again. So, and I wasn't able to say, show me my QR code for my valid guy. There is an option to get the equivalent 26-character key. And so because what I wanted was I wanted to try this also on my BlackBerry. So I had it on my iPad and my BlackBerry. Now, the BlackBerry Google Authenticator is bare bones, to say the least. I'm glad to have one, but there's no frills. There's no option to have it snap it with the camera. You have to enter it by hand. And as soon as you respond to the QR code, then you no longer have the option to get the 26-character TOTP key. So I had to delete it and start over. This time I got the 26-character key, so I could write that down. Now I have that for my Dropbox account so that I'm able to manually provide it to any instances of these TOTP authenticators that I want.

So to me it feels like it's a little bit awkward yet. It'd be nice, I mean, it's great that they have it. It works. I really think this is going to be the future. It looks to me like this is the two-factor authentication technology which is - it costs nothing. It's multi-account inherently. It's standards-based. It's secure. I bet you we're going to start seeing it built into devices. It's the right solution.

Oh, and one last thing, they do provide, Dropbox does, a 16-digit backup code which they give you as your override. So if you - this is something that you write down and put in the safe or in your safety deposit or somewhere because, if you can no longer do your two-step verification, I mean, arguably there has to be some solution for convincing them that you're still you. So there's a much more cumbersome, yet reasonable, super-secure because it's 16 random digits, actually I think it's characters, I can't remember whether mine - I think mine was full alpha. So very, very high entropy code they provide which allows you to get back in if for some reason you can't do so with your one-time password device. So I feel like the UI needs a little bit of work. But we now have two-factor authentication for Dropbox. So that's definitely a yay.

Leo: I don't know if this is the one Dropbox is recommending. There's a third-party authenticator that works with Google 2-step. So I presume this is what they're talking about for Windows Phone. But it's from a company called Slug on a Mission. I, you know...

Steve: Is there something - the word I have is just "Authenticator," as if that's the name.

Leo: That's what this is called. But golly. I would rather it were offered by, I don't know, somebody besides Slug on a Mission.

Steve: And then they also mentioned Amazon AWS MFA, that's probably Multi-Factor

Authentication, for Android. Although Google Authenticator works for Android. I don't know why you'd look any further than that. So anyway, the idea is, if I had that football, and I still do, and I use it, I don't have control of it generally. That is to say, PayPal uses it with VIP, presumably, VeriSign Identity Protection, paying the price to have that kind of protection. But it's not generally available. I mean I can't use that for other things unless those services sign up to validate with the same back end.

The beauty of this transformation that we have now seen, probably led largely by the presence of the Google Authenticator, is the concept of let's take the time-based one-time password and allow the user to provide the key, and we'll use that cryptographically to generate the code. And we can have, the user can have as many accounts as they choose. So there will be Dropbox, there'll be Google Mail, there'll be Amazon S3, whatever. And I think that's probably going to be the solution that wins for this kind of multifactor need. So I'm glad for it. I think we're going to see...

Leo: Yes. Everyone should go two-factor. Everyone.

Steve: Yes, yeah. I got a tweet from James L. McMahan, Jr. that I just thought I wanted to thank him for, showing that Revision3 is detecting ad-blockers. And when he went there with his ad-blocker on to Revision3, it said - he got a little note on the screen that said, "Oops, your ad-blocker is on. Revision3 content requires ad-blocking software to be disabled. Thank you for your support." And I think that's entirely appropriate, Leo. I mean, I think that's the right thing to do. We've talked about the inherent tension that exists between tracking and advertising and all that. And I have absolutely no problem with the idea of a site saying, wait a minute, you're not accepting some content which we need you to accept in order for us to be able to give you every thing else we want to. So please accept it.

Leo: Good. And I accept it. We don't do that, by the way. But I understand why they do that. They've got to pay for their bandwidth and stuff.

Steve: Oh, yeah.

Leo: That's not free. I wonder if that's a Discovery thing, now that they're owned by Discovery. I wonder if Discovery does that.

Steve: Oh, interesting.

Leo: Yeah.

Steve: I tweeted, I'm trying to think, late last week I think, about something that I had received a lot of tweets about previously. It's a really nice directory of free online Internet courses, Coursera.org. Coursera.org/courses is a list of all the courses that they have available. And just, I think it was yesterday or maybe Monday, the Crypto course, which has run several times, was restarting. So that was Coursera.org/course/crypto. And I got a lot of people who thanked me for the tweet. Some signed up. Some said, hey, I did that last cycle, and it was really good. One guy, in fact, who's a listener,

obviously, said, hey, you'll get a kick out of the fact that the MS-PPTP, which is what of course we talked about as not being secure last week, he said, that was used as an example of an insecure protocol in the course. And many people have said that this stuff, especially maybe the crypto course, is very good. So I just wanted to bring it to our listeners' attention, Coursera.org/courses.

Leo: It's very cool. It's very cool.

Steve: Yeah, and these are major universities. I mean, this is Stanford and Princeton and Columbia and Rutgers and a bunch of real players who are doing this. And total miscellanea, but I missed the 80th birthday of LEGOs a couple weeks ago.

Leo: Man.

Steve: And that just - I saw that today, I said, oh, LEGOs turned 80. Wow, that's very cool. LEGOs, of course, are a favorite geek toy.

Leo: And no longer copyrightable or patentable.

Steve: [Gasping]

Leo: So anybody can make - that just happened in the last year or so.

Steve: No kidding.

Leo: For a long time you could not make a LEGO-compatible block. And now you can.

Steve: Wow. I wonder how they kept the intellectual property for that for 80 years?

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

Steve: Yeah. I did get a nice short note from a happy SpinRite user, Mark Cole, who wrote back to Sue. He must have been corresponding with her about something, oh, apparently about our consultant license. He said, "Sue, thank you for your prompt reply and thank you for the explanation. I'm sorry I missed the specific web page you referred me to, but I am so glad you have consultant licenses. I'll work towards purchasing the four copies."

Just to pause for a second, the way we do this is you buy one copy of SpinRite, and you can use it on all the machines you yourself personally own. But of course there's been a demand over the years for people who are computer fixers to be able to use it on all of their clients' machines. And so we said, well, if you keep four copies current, then you're

entitled to do that. That seems, I mean, nobody else is - I just kind of invented that. I like it, though, because it allows someone to try one copy, and then they don't have to, like, ask for a refund for their one in order to get a consultant license or something. I mean, it's just, it's like, oh, no, just get three more. And so I always smile when I hear three yabba-dabbas come out of our eCommerce system because that tells me, first of all, that someone tried their one copy, and it worked, and they liked it, and also that they're being honest, actually...

[Talking simultaneously]

Steve: They're saying, hey, you know, I'm going to get three more so that I am a valid consultant and can use it on all of my friends' machines.

Leo: Aren't they nice.

Steve: So anyway, so he finished, says, "Also I wanted to share that I went to the location where I was working on the PC with the Blue Screen of Death, and SpinRite comes to the rescue again. It took a couple of reboots after SpinRite did its thing, and Windows XP followed up with doing its own chkdsk, and the PC is up and running like nothing ever happened. The customer is going to be absolutely thrilled when they come in tomorrow morning and their PC will be up and running. Thank you. Mark Cole." So, Mark, if you're listening, thank you.

Leo: Winna winna chicken dinna. I like it. And now we move on.

Steve: Okay. So two good TNO, that is to say Trust No One, cloud storage clients or solutions. And they're very different, but I wanted to bring them to our listeners' attention. I've been playing with them both. And one of them I think is going to end up being the official Security Now! solution. But I'll talk about that one in a second.

First is a very lightweight little, I almost want to call it "applet" or "utility." It's called DataLocker from AppSense Labs: appsense.com/labs/data-locker. And I just tried to Google DataLocker, and I can't tell whether it's the same - I don't think that's the one that comes up. Oh, it is the second link on Google, if you just put in "DataLocker" into Google. The second link you'll see www.appsense.com/labs/data-locker. Anyway, this is - it's very small. It's one file. It's 1.576MB. It does require that you have .NET v4. And I'm thinking that it is cross-platform, but I didn't write that down. I think there's a way of running similar apps over on the Mac with some library that you need to have.

Leo: Yeah, yeah. There's an open source .NET. Maybe that's how you do it, yeah.

Steve: I think that's it. And so, okay, so this is minimal, minimal, minimal. And I like it because I like small, lightweight solutions. It's simply a drag-and-drop target, so you drop a file onto it in a little window that it provides, and it asks you for a password, which is not recoverable, which it doesn't store. You put the password in. And then by default it encrypts it, adding its own extension to the end, and returns it to the same directory that the file originated from. Or you can send it to a different directory of your choice. And there's two tabs. There's Encrypt, and there's Decrypt. And that's all it does. But it's cute.

Now, they're not apparently taking it very seriously. It feels to me like it's something they did to draw attention to themselves, and here it's working, because they are, like, all about other things. I wrote to them a couple days ago and said, hey, this looks very nice, but there's no documentation anywhere about the crypto protocols you're using, the file format or anything. So what can you tell me about that? Silence. Never got a reply from them. Other people have said they didn't get a reply. And I told them that we have a podcast, and lots of people, and this might be interesting, but I don't know. Maybe they're not looking at that email anyway. To me it feels like it's not a mainstream deal for them, but something, I mean, it's so, frankly, simple and easy to do this.

But it's cool. It's a little drag-and-drop simple encrypter/decrypter that you give a filename to. It doesn't remember the password. I don't know how I feel about that. It might be nicer if it was sticky. It doesn't remember the directory that you tell it to put things in, and it would be nice if that was sticky so that, for example, you could use it with Dropbox or one of those utilities, cloud services. But anyway, it's a cute little, I mean, to me, I imagine they're using AES-256. I looked at the encrypted file a little bit, and there's a little bit of a header that they add to identify themselves. I have no reason to believe that they did anything nefarious. And it is very small and lightweight. So easy, quick and easy drag-and-drop encryption.

Now, the one I'm impressed by is called Duplicati.com. It is over on code.google.com, so open source, being run by a couple guys. And I'm, well, let me just tell you what it does. It is a very flexible general-purpose TNO encryption backup solution that is completely oriented toward cloud usage. It is Windows, Mac, and Linux, so all three major platforms. It has awareness, specific awareness of the Amazon S3 service, Microsoft's SkyDrive, Google Drive/Google Docs, Rackspace Cloud Files. It can also just go - it can operate on a file-based basis. It can talk to WebDAV servers. It understands the Tahoe-LAFS, the Least Authority File System, which is a distributed secure cloud file system. It can also talk to FTP servers and also SFTP, Secure FTP, using SSH. So very flexible from that standpoint. AES-256 crypto.

It will do a full backup of chosen subdirectories and then follow that with incremental backups, and you can tell it how long you want it to do incremental backups before it does another full backup, so it just doesn't do that forever. There's a command line tool available also for it, for power users, although it's got a nice little UI. And I'm trying to think how much storage it uses. It's been running on my machine for, like, a week or two. And it looks like it's 45MB, so it's behaving itself well.

There's something in Windows called the Volume Snapshot Service, VSS, which allows backup utilities to perform backups of files that are open. This has traditionally been a problem, for example, with people who just run with Outlook open because the PST, that central Outlook database file, causes backup problems for utilities because it's open all the time. So it's able to do - it uses the VSS under Windows, or the equivalent, the Logical Volume Manager under Linux. Very nice and flexible control over what it backs up.

I sent my entire source code tree under my assembly language directory, which of course I don't do casually. I sent it up to Amazon S3. I was able to say I want to back up this entire tree except exclude .EXEs, .OBJs, .RESs, basically all of the intermediate files other than my include files, my header files, and assembly code files and so forth. So you're able to say I want - very accurately specify. It's got a nice sequential process rule-based system that understands regular expressions or simple expressions. Internally they're all regular expressions. So, I mean, like, really lots of control. So you can say match on this, then exclude if this, and then include if that, and so forth. And it's not huge. The entire directory containing it, it's got lots of little itty-bitty parts and DLLs and things, but it's

about 18MB installed. And again, cross-platform, cross-service.

I'm using it, and I will report back after a while. But I wanted to bring it to everyone's attention. I am very impressed. It's a nice little system. The idea would be, then, that everybody's offering free storage now, Dropbox and Google and SkyDrive and so forth. This also allows you to have different named backups, which you can collect in groups and control how often and when and under what circumstances they are run. So you can have multiple groups with multiple backups. The backups can contain multiple folders, subdirectories, and those can have the whole rule-based system applied. So it's very hierarchical and tons of control. And each group can be sent to a different service. So it's not like you have to commit the whole thing to Amazon or to SkyDrive. You can say, okay, I've got a bunch of free space scattered around. I want to put these files over on this service, and those files over on that service. And this thing just does the whole thing. It's a terrific little gizmo.

Leo: Too bad no Mac version of it, unfortunately.

Steve: Yeah, there is. Windows, Mac, and Linux. Yeah, all three platforms.

Leo: I saw the Windows and Linux, Linux using Mono. I guess if you use Mono on the Mac you could do it.

Steve: Then you're ahead of me. I did not pursue it beyond...

Leo: Yeah, I see it, I see it. Released on, yeah, yeah.

Steve: So Duplicati.com.

Leo: You have mentioned this before.

Steve: Did I?

Leo: Yeah.

Steve: I guess I didn't drill down into it and look at it closely.

Leo: Yeah, I think it was in your long...

Steve: My big roundup?

Leo: ...roundup. But so it's something that - I know I've seen this before. I'm still using Arq, which you had recommended on the Mac, which I really like.

Steve: Yup. And this is also full TNO. So they're doing crypto right.

Leo: Yup, and this is free.

Steve: So, Leo?

Leo: Yes? Yes, Steve?

Steve: The patent system is messed up.

Leo: Yeah, well, I agree with that.

Steve: And I've been watching and listening to people talking about it for a long time. One of the problems, one of I think the clearest problems is that 20 years is a long time to provide protection for things that don't involve, like, the building of factories and laboratories and facilities in order to implement the patent. Software moves very quickly. And granting really broad rights to a company which largely uses its size in order to enforce and in some cases intimidate is discouraging.

I wanted to tell our listeners about a site that the EFF has put together called DefendInnovation.org. They have a petition there, and they're collecting names and just a little bit of information to assure the world that you're legitimate. And I would invite people to go over to DefendInnovation.org and take a look around. They break their concerns down to seven points. The first one is that a patent covering software should be shorter. And they're suggesting no more than...

Leo: It's 17 to 20 years right now.

Steve: It is, exactly.

Leo: Which is ridiculous. I think it shouldn't - they even say five years, but I think maybe five months or a year because cell phones move pretty darn fast.

Steve: Right. And, I mean, as someone who has experimented and played with the patent system, I've written patents and have attorneys, and I've explored this, there are many problems with the system. One of the problems, I think the most pervasive problem is that it's often just the company that looks at something first, that basically says, oh - like Apple did. And I'm an iPhone user, iPad user. I love Apple stuff. It's beautiful. But the tendency is to get patents on everything, even if somebody else who was put in a clean room, had no contact ever, and asked to solve the problem, would just

do the same thing. I mean, oftentimes things are obvious.

And the obviousness test I think is one of the biggest problems we have because, unfortunately, the way our legal system works, we get a jury of 12 people who were selected randomly from the voter roles and the DMV, and is this obvious to them? No. But it's obvious to every other programmer on the planet. But you didn't get 12 programmers on the jury. So they say also, if the patent is invalid, or there's no infringement - and then here's a little bias showing through. They say, "The trolls should have to pay the legal fees."

Leo: They're not talking about Samsung here. They're talking about people like Nathan Myhrvold who collect patents purely for the point of either extorting license fees or sue.

Steve: Just to litigate, exactly.

Leo: Yeah. The problem is, that doesn't distinguish between people like Apple and people like patent trolls. So that's kind of the challenge here is Apple, I mean, regardless of how you feel about the outcome, Apple acted in good faith in suing them.

Steve: Oh, absolutely. And...

Leo: So had they lost, having them pay costs wouldn't necessarily be appropriate.

Steve: Yes. And, for example, I remember smiling when I looked at my - I was playing with my Nexus 7 tablet. And when I come to the end of something, it doesn't bounce.

Leo: Right.

Steve: It stops.

Leo: For a good reason. That's a patent.

Steve: Apple has a patent on bouncing. And it's like, well, okay. I mean, so, I guess - and we've talked about this in various contexts before. I love the fact that I can hold the button down on my BlackBerry, and the letter capitalizes. Well, I can't do that on my iPad. It would be nice if I could do it on my iPad. But BlackBerry has that, and so Apple can't offer that. So, I mean, there's a problem here in that there's certainly a tension between features that would benefit the consumer and competitive features that benefit the companies offering them. And so to your point, Leo, should that be 17 years?

Leo: No, it's too long. It's too long.

Steve: It's crazy, yeah. And so it would be really nice if, for example, if hold button capitalization, which BlackBerry came up with first because they had a phone with a keyboard on it before Apple ever looked at the situation - now, again, it's a matter of who got there, who had the problem first.

Leo: But this is kind of the point of FRAND [Fair, Reasonable, And Non-Discriminatory], which is fair-priced licensing should be required. And a lot of companies are agreeable to FRAND.

Steve: Yes.

Leo: And that maybe is what you want to enforce, which is, yeah, sure, somebody deserves a license fee. But it should be a reasonable license fee. You know Apple was asking $30 per phone. Whether that's reasonable, I don't know, but it was so much that no company could afford it.

Steve: Right, right. And so on the issue of trolling, for example, the EFF said: "Shift court fees away from innocent parties: Both the winner and loser in a patent suit almost always pay their owns fees and costs, which can total well into the millions of dollars if the case actually goes to trial. Because the potential costs are so high, and there is no way to recover those costs, defendants will often settle to avoid hefty legal bills - even if they have a strong legal case that they never infringed on the patent or the patent was invalid to begin with." So...

Leo: Right. That's the case in lawsuits in some states. They have a SLAPP law, which means if it's deemed a frivolous lawsuit, you pay the costs if you lose.

Steve: Yeah. And again, it's not, I mean, frivolous is a value judgment. I would imagine that most patent suits are regarded as serious suits. But anyway, so the point is, at this point, these suits are so expensive that somebody in the right is in tremendous peril and expense if they push this. So it is, I mean, one of the complaints that I've had generically about the patent system is that, because the patents are complex, the Patent Office, the theory is that the examiners are, I don't know, Einsteins - actually, that's a bad example because he was a patent examiner - but the idea that the patent examiners are, like, omniscient and actually only issue patents for inventions. But they're not. The Patent Office is overwhelmed, understaffed, underfunded. And so what happens is patents get granted if it's not obvious that they should not be granted. And they just figure, well, we'll let the courts battle it out. We'll let the courts figure it out.

Leo: Right. And that's what happened, I think, with these patents, in fact. It's like, well, it looks like a good patent. We'll just make it - we'll prove it, and we'll see what happens.

Steve: Right.

Leo: I don't think people understand that, that in fact it's often the courts that are asked to rule. And that's inappropriate.

Steve: Right. The fact that you have a patent means nothing. And in fact, I'm sure you're probably - you'll be able to pull up the number or the name of this. But there have been sort of some crowd-sourced efforts to help with prior art because the way the patent system works, and this was the instance of some dialogue that you captured at the beginning of the show, before we began recording, Leo, where the bore on the way the Apple vs. Samsung suit went, the idea being that oftentimes there'll be a patent that's been granted for something that already existed in some corner somewhere, in some university, some grad student published this. And in completely good faith the person submitting the patent may have been unaware of that. It is the case that the same stuff is independently invented all the time.

Leo: Sure.

Steve: I mean, it's the nature of this. I mean, and that's one of my problems with this notion that the way this really ends up happening is because this concept, the definition of what is an invention has been so weakened and watered down that it's just who was - what engineer was asked to solve the problem first. Now, I mean, I love Apple's springy pages, and the pinch-and-zoom thing. But it's like, oh, okay, was that an invention?

Leo: I don't think you can argue that the thing, the patents Apple was suing over were for things that made the iPhone better than other phones. You don't need the bounce. The Nexus 7 lives without it. The tap to zoom is great. Apple invented it. There's no question. Although they did show prior art. It was actually interesting because I played this cut before the show began for you. The foreman of the jury, in an interview, said he had an insight about one of the patents because he has a patent of his own, so he's an expert.

Steve: Ah.

Leo: I think that's why he was elected to be foreman, I'm sure. His patent, by the way, as far as I could tell, is completely generic and meaningless.

Steve: I'm surprised that he got past the attorneys.

Leo: I am. I am, too, because that's the guy that the jury is going to end up looking to. Hey, you have a patent, you understand this process. And he, now, I'm very curious about this because he says, I was really struggling with this particular patent. I think it's the tap-to-zoom. I'm not sure. I'll have to go back and look. But I had an epiphany. I went home, and I had an aha moment. Apple's invention wouldn't run on the prior art device that Samsung submitted into evidence, nor would Samsung's code run on Apple's device. Different processors. So it wasn't, he says, he used this to determine it wasn't prior art because it didn't run on the same

processor. That doesn't make sense. Is that right?

Steve: No, no, that's a complete misinterpretation of prior art.

Leo: That's what scared me. And because he's the expert, the jury bought it.

Steve: If Apple had been patenting the particular algorithm code, the algorithm that they used, then that would be one thing. But they're not. They were patenting bounciness. Just like - or pinchiness. I mean, they were patenting the concept of how can we gently tell the user they reached the end? Now, I've got to say, Leo, I like that enough - and I guess I'm supporting the notion that this is valuable to Apple. I like that enough that, if I could pay Apple 30 bucks to add the bouncy package to my Nexus 7, I would probably do it.

Leo: But this was Apple's contention. We made some inventions that made our products better, and Samsung just blatantly copied them so that their products would be as good as ours. But in fact we deserve the right to have this uniquely for 17 years. And I don't, I mean...

Steve: 17 years.

Leo: There's problems. I understand that.

Steve: We're going to be beaming ourselves up to another planet.

Leo: Right, no, I agree. And that's - but Apple was acting with the current state of the law.

Steve: Oh, yeah. I mean, they're acting in the interests of their shareholders, which they're obligated to do.

Leo: Right.

Steve: And I guess my point is that, as a consumer, I can't have "hold the button down and capitalize it" on the iPad.

Leo: I know, it drives me crazy, right.

Steve: I have it on the BlackBerry. I can't have bouncy pages on my Nexus 7. I want them.

Leo: I agree.

Steve: And so it would be nice, for example, if it were possible to, like, purchase these accessory patents.

Leo: Can I license them from Apple.

Steve: Right.

Leo: Let me license them.

Steve: Yeah, yeah.

Leo: It's a real - it's a thorny problem. I think - I signed the petition because I do think we need patent reform. DefendInnovation.org.

Steve: Well, yes, let's just revisit this. Let's get some smart people, I mean, and you see me, I'm not saying that that was right or wrong. I love the bouncy pages. I understand that that's valuable. I would argue a little bit about whether somebody else encountering the problem wouldn't have the same solution. So do you know what I mean?

Leo: It's the obviousness of the patent, right.

Steve: Yes. They just happened to get there first.

Leo: And the Patent Office is, I think, supposed to consider the obviousness.

Steve: Oh, yeah. The way the actual language is, anything is disqualified for being considered an invention if it would be obvious to someone trained in the art.

Leo: Right.

Steve: That is, if somebody, if a software engineer were given the problem or shown this, would it be, like, obvious? Or would it be like Velcro, oh, my god, or a zipper, maybe, which I still can't figure that out.

Leo: Yeah, zipper is not obvious. I have no idea how that works. I understand Velcro. That is why, in the Google/Oracle case, Oracle lost its case because one of the patents that they were fighting over was a range check. And the judge even

said, no, this is obvious. I learned Java and wrote a range check last night. It isn't, it is not a non-obvious thing. So the judge threw it out. There was no jury in that one. And I have a feeling, had there been a jury, it might have been a different outcome.

Steve: Well, as I was telling you before we began recording, Leo, a couple decades ago I used to accept assignments or opportunities, whatever you would call it - consulting, I guess - as an expert witness in trials. And I was involved in several that were intellectual property trials. And when contacted by the attorneys, they'd say, hey, Steve, you wrote a column in InfoWorld where you said the following things, and we agree with you, and we have some litigation about that. And so I would understand what it was they were talking about, and if I was on their side, then I would say, okay, I agree with you, so this sounds interesting. The problem was I watched the court system just fumble over and over and over, I mean, essentially reaching what I knew was the wrong conclusion. So finally I said, okay, I'm not doing this anymore. It's just too frustrating. Oh.

Leo: And I have to say politics and the law and courts are frustrating for normal people. Thank goodness there are people who have longer attention spans than you or I.

Steve: The news was that it was a $1.something billion judgment. And I guess that it was willful infringement.

Leo: Which means the judge has the right to triple it because it was deemed willful.

Steve: Ohhh, ow.

Leo: Now, we don't know what she'll do. And I think the case can be made, frankly, that Samsung, it's pretty clear from the evidence, this is what the real - really the jury did rule from the Samsung emails in which they said we have a crisis of design, we've got to do something, this iPhone's too good. They kind of - there was a smoking gun.

Steve: They copied it.

Leo: It's pretty obvious that Samsung said we've got to copy it. And I think it's reasonable to say that Samsung did a calculation and said, look, the best business would be at this point to copy it, take the chance, pay the fine if we get fined. They probably talked to Apple. Apple said we want 30 bucks a phone. Samsung said there's not enough money in the phone to give you 30 bucks a phone, so we're not going to pay that, we'll just take our chances. And they lost in court. They made $21 billion on these phones. They'll pay the $1 billion fine or even a $3 billion fine and say that was our R&D cost for these phones. And now...

Steve: And they'll take the bouncy out of the...

Leo: And now what they're doing is they're responding and they're changing and they're innovating. And you look at the phone that Samsung just announced, which is the Galaxy Note II, it's not an iPhone. It's 5.5". So there's no question. Nobody's going to look at that and say, oh, yeah, you stole the iPhone. So Samsung I think probably made a very conscious business decision infringe. That would be my call.

Steve: Well, and Samsung is also still the huge major part supplier for Apple for all of these phones that...

Leo: Yeah, they're frenemies.

Steve: ...it's been sued against copying.

Leo: Yeah. And that business never went away, by the way. At no point did Apple say, oh, we're going to go somewhere else.

Steve: Yeah, there is nowhere else.

Leo: Well, there are. There's LG. There's other people they could go to. It's a big - there's a big business here.

Steve: So I would ask people to go, consider taking a look at that site, DefendInnovation.org, sign the petition if you think as we do that this ought to get some attention, that 17 to 20 years is a long time for a company to own exclusive rights to something that an engineer said, oh, let's do this. A few years would be a good thing. And then - or it'd be great if, like, reasonable licensing terms or, who knows, something, I mean, it'd be...

Leo: I suspect that we will, in fact - this has been such a high, I mean, these suits go on all the time, and there are many more. Motorola's suing Apple. There's plenty more going on. And I suspect this is such a high-profile case that these companies are going to say, look, this is a war of attrition, nobody's going to win, let's figure this out. And it's actually over now. This was only a crisis in the first few years of iPhone dominance. I think it's pretty much over. I think there's a - the two have gone their separate ways. No one's going to confuse an Android phone with an iPhone.

Steve: No.

Leo: It was a good show, Steve. Very interesting stuff. It was a busy week. Now, we will do the Q&A next week, so we're back on our Mod 1 or Mod 0, whatever it is.

Steve: Yes, I was thinking, well, we have the advantage of skipping back to the original

phase that we were in. We had a 180-degree phase shift, and we've phased it again.

Leo: It's Mod 2, right.

Steve: We're back in phase, yes.

Leo: It's even-numbered shows. So next week, if you have a question, a comment, or something you'd like to say and have Steve address it, you can go to GRC.com/feedback. That's where - don't email me, don't email Steve, just the feedback form is meant for this purpose, GRC.com/feedback.

Steve: Yeah, I'm nodding, but no one can see that.

Leo: Steve is nodding.

Steve: Yes, I'm nodding. So, yes, go there.

Leo: Yeah, nodding doesn't work on audio. We do make audio and video versions available of the show. I don't know why you'd watch the video except to see Steve's smiling face and his nods. But you can get both at TWiT.tv/sn. And of course Steve makes the little teeny-weeny 16Kb version available on his site, as well as full text transcripts by a human being, which means they're legible and spelled properly. That's at GRC.com, where you also find SpinRite.

Steve: Oh, and I actually forgot to give Elaine credit for the Coursera.

Leo: That was her pick.

Steve: Several people had mentioned it to me, but I just hadn't gotten off the dime to go pursue it because I would see the tweets come in when I was in the middle of something else. And finally Elaine said, hey, I was - the thing that made me think about it is that she does so much research when she's doing the transcriptions. She's going out and checking spelling and verifying things, I mean, there's a whole lot of - it's much more than just an automated process for her. And so something that she was doing took her to that site, and she said, hey, I'll bet this would be interesting for Security Now!'s audience.

Leo: Coursera's amazing.

Steve: So thanks to Elaine, yeah.

Leo: I just wish I were a young person again, and I had more time.

Steve: Oh, gosh, I know.

Leo: SpinRite is also at GRC.com. That's Steve's bread and butter, the world's best hard drive maintenance utility. You must have a copy and get it there. Also lots of freebies including ShieldsUP! and the Password Haystacks, all that stuff. GRC.com. We do this show every Wednesday, 11:00 a.m. Pacific, 2:00 p.m. Eastern time on TWiT.tv. That's 1800 UTC. Now, Steve, just a program note, I think we mentioned this last week, two weeks from now, supposedly, although the invitations haven't gone out, is Apple's...

Steve: We're going to do a Tuesday/Wednesday swap?

Leo: Yeah, we might. We don't know. But I suspect we'll want to do a Tuesday/Wednesday swap on Episode 369. But we'll let you and everybody else know before we do that because of the Apple announcement.

Steve: And that's supposed to be the iPhone 5; right?

Leo: That's going to be the new iPhone. I think there seems such a consensus among all the pundits that I think we're pretty much sure that that's what will happen September 12th.

Steve: And a 7" tablet has been pretty much confirmed, too.

Leo: But not for that - now the consensus is that it will be a separate announcement in October.

Steve: Right.

Leo: So, you know, it's all made up.

Steve: I like mine at 10.whatever it is. That's, to me, I'd like, if I think about the same thing in a smaller one, it's like, I don't know.

Leo: I'm excited. I love the 7. You have the 7, the Nexus 7. And I love that size.

Steve: Oh, I do. It's beautiful.

Leo: But I'm excited about the Note II, which is 5.5".

Steve: Okay.

Leo: I think that that's going to be a perfect in-between because it's a phone and it's a tablet.

Steve: And a stylus; right? A stylus-based tablet.

Leo: It has a stylus. And it's actually the stylus that it uses is very sophisticated. You can do some really interesting things.

Steve: When is that happening?

Leo: They didn't say a date, but I think it's generally thought that it will be available in Europe next month. The European versions always want to be ahead of everybody. And the American carriers will get it in December.

Steve: Ooh, December, that's a little late for Christmas.

Leo: Before the end of the year.

Steve: Ah.

Leo: Steve Gibson, thank you so much. We'll see you next week on Security Now!.