SECURITY NOW!

Transcript of Episode #246

Listener Feedback #91

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-246.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-246-lq.mp3

Leo Laporte: This is Security Now! with Steve Gibson, Episode 246 for April 29, 2010: Your questions, Steve's answers #91.

It's time for Security Now!, the show that covers everything you would ever want to know about protecting yourself online, about your online privacy. Boy, that's a big topic these days. And the guy who knows all, who sees all, who tells all, Mr. Steve Gibson from GRC.com, the creator of SpinRite, the world's best hard drive maintenance utility. He's also creator of a number of really useful security utilities and joins us every week for the last 245 weeks to talk about security.

Steve Gibson: Yeah, in fact, I looked at the number, and I though, whoa, wait a minute, where are we at our five-year mark? And that's when we're at Episode 260.

Leo: We're close.

Steve: Today is 246, so we're 14 weeks away from finishing year five. So...

Leo: Now I'm tired. I wasn't tired when I started. Now I'm tired. That's a lot - that's amazing. But, by the way, folks, and this was I think always Steve's goal, this archive of 245, now 246 episodes is really a great primer in security, how it works, how computers work, what cryptography is, how - what the principles of it are. And I think that was always your goal is to create this body of knowledge that people can refer back to that is so useful. So please take advantage of that.

Steve: Yes, when you and I were first talking about it, my first thought was, well, that this could be a long-term collection rather than something transient, very much like what I did with you when I would pop onto the Tech TV shows, trying to do sort of useful foundational stuff. And what I think is surprising is that that kind of material has an extremely long life. I mean, we're talking often about fundamental aspects of technology that just don't change all that fast. I mean, new stuff comes along, and we talk about that, too. But as you say, it's a great repository.

Leo: Well, you've done a good job, and we're going to keep doing so. This is a Q&A episode. That means you've got questions and answers from your audience, and we're going to get to those. We also have some really interesting security news.

Steve: Yup.

Leo: Today I decided to deactivate my Facebook accounts because I'm so concerned about the privacy issues that Facebook is raising with this new Open Graph initiative. And I'm just not understanding how much of my information is leaking. And you know what's a little troubling, you can opt out on Facebook from sharing your information, but then there's this little troubling statement that says, well, you've opted out, but you know your friends can still share your information with other people. And you have to opt out of that, too. And the whole thing, the structure of it is too complex for me. I'm very concerned about it. So I know you don't have a Facebook page.

Steve: I don't. Well, but I have, well, I mean, I was going to say I have a website, which is sort of my...

Leo: That's different. And so that's what I'm going to do.

Steve: ...equivalent, yeah.

Leo: Yeah. That's what I'm going to do.

Steve: It's funny, there's just a, in reading through the mail this morning, choosing questions, there was one that I didn't choose, but a person I responded to, who was just sort of talking about the philosophy of security. And I mentioned that, for example, I will - I just have these instincts built in now. I will talk about trips that I have had, like when I have returned. But I never talk ahead of time about a trip I'm going to take. It's just built into me. This information gets out there, and I've got a wide audience, and it's just like, okay, I'm happy to talk about where I was. I don't want to talk about where I'm going because it means where I'll be, and it also implies where I'm not.

Leo: Yeah, I have to get to that point. And I've been a little open. And before I get bit I should probably reconsider that whole thing. It's a little different because I'm a public figure, and this business relies on me being a public figure. And it's something

I signed up for, in effect. But boy, you do have to really worry about that.

Steve: It's just sort of a mindset.

Leo: Yeah. All right, Steve Gibson. Lots of security news. Let's get right into the matter here.

Steve: Yeah. First off, it became - I guess it wasn't an official announcement, but somebody who said he really wasn't supposed to tell what was going on did reveal to The New York Times, and I saw it somewhere else, that Google, as a result of these attacks that Google had suffered, which we've talked about on a number of occasions, they've continued to go back forensically and look at what it was exactly that happened. And they now know that they lost control of a chunk of their login management. They call it Gaia, G-a-i-a, which is their so-called Google single sign-on. And they lost the source code, meaning it wasn't stolen from them, but somebody who penetrated their network was able to get a copy of it.

Leo: Holy cow.

Steve: And so this is the way, for example, that you log into sort of the Googleplex as, like, using your Gmail login, and then all of the various services you're simultaneously logged into. And that's what we mean when we say "single sign-on." You just - you authenticate yourself to Google once, and that creates some persistence throughout their services. So what we now know from this leak that is on the inside, is that a Google employee located in China received a message through Microsoft Messenger containing, we know, a link.

Leo: Oh, boy.

Steve: The employee, the Google employee in China clicked on the link, which was delivered by Microsoft Messenger, which linked to a malicious website which then installed a trojan. It infected that employee's machine, installed a trojan that gave unknown parties access into Google's network. Using that machine as a launching point, they were then able to penetrate the network and get to the software developers' source code repository where the single sign-on code was stored. And one of the security researchers, actually with a different company, made the comment that, if there was as part of the source code repository an internal list of known problems with the current single sign-on system...

Leo: Of course. The bug list.

Steve: Exactly, like a list of things that they intend to fix or they're working on, then that could be extra problematical. And of course the danger is that, given the source code, bad guys could go through it with an eye toward finding opportunities for exploitation that it's very difficult for Google's own people to see. I mean, we've talked often about this, how odd this mindset is. It's one thing to look at source code and say, is it going to protect people from, well, is it going to provide secure sign-on services, which is what the authors of this system want, and it's just - it's such a different mindset to look at it with an eye toward how can we maliciously exploit the same code? What can we feed this that will cause it to react in a way that the developers didn't want, but which was behavior that just got included by mistake? And of course that's what the bad guys will do.

So, I mean, this doesn't mean anything necessarily bad, except it was in the news this week, and I thought our listeners would find it interesting. And, I mean, potentially, potentially only, it's of concern. So for me this demonstrates the value of maintaining logs, which must be the way this kind of forensic work is being done, is that there is lots of logging being done so that researchers are able to go back through and determine where the penetration came from, backtrack that to the machine, backtrack that to a Microsoft Messenger message received by this particular employee on this machine that clicked this link, and that's how it happened.

Leo: Wow.

Steve: I mean, that's a amazing forensics. But that's what Google now knows about at least one aspect of the penetration into their network.

Leo: It's a little different now because in the days of, you know, like "Cuckoo's Nest," Cliff Stoll and so forth, the first thing a hacker who broke in would do is modify the log so that there was no trace of the hacker, or at least attempt to. But with these scripts and these trojans, you don't have access to the logs necessarily. You're not necessarily rooting the machine, for instance.

Steve: Precisely. In fact, exactly. In the instance you're talking about, you'd be getting onto a specific machine, and you'd be changing the logs on that machine.

Leo: You'd have to have root access to do that, of course.

Steve: Well, and here we're probably talking about logging servers on entirely different...

Leo: Different machines, yeah.

Steve: ...networks or different machines that are logging traffic. And so traffic logging is much different than local machine activity logging, which is, in old school, that's what UNIX machines were really good at doing, rotating their logs and compressing the logs and keeping this in order to determine, if something went wrong, what was the cause for it.

Leo: Right. Fascinating.

Steve: In another very different story, there was news actually from last week that the cellular GSM system was legally hacked by a pair of security researchers, Nick DePetrillo and Don Bailey. And what I mean when I say "legally hacked" is they very cleverly took advantage of just the globalness, essentially, of GSM, and features the system has to incorporate in order to do what it does. The first thing they did was to realize that the caller ID system has an API, meaning that in order for a phone to have - in order for the GSM system to support caller ID, there's got to be essentially an open directory which is able to map phone numbers to owners.

Leo: Right.

Steve: So, I mean, that's what caller ID is, is it says this phone is calling you. Oh, and by the way, here's who it is. And in one article that I read they used Brad Pitt as an example just of someone famous. And so the idea is, what these guys realized was that they could set up - they used an open source PBX. I had the feeling it was Asterisk, but I'm not sure. They used an open source PBX, but something like that, where the API is supported. And they're able to walk the tree of all possible cellular phone numbers because we know what the prefixes are on those. And so they basically just query this globally available API to build their own dictionary, their own mapping, basically, suck this database out which is a distributed database, pull it all together into one place, which maps phone numbers to people's identities.

Well, then the other thing that has to be possible for a global cellular network to function is there has to be inherently a location system. That is, it must be the case that, when you want to call some phone number, a GSM phone number, that there's a way for a network where you are to know how to route the call to the network where that destination phone number currently is, that is, its current location. And so they worked, they reverse engineered that and figured out how, essentially, to determine the location, that is, in terms of, like, city and even subcity geographic proximity for any given phone number on the GSM system. Which of course somewhere in the system that has to be available in order for you to send - in order for the system to find the phone, wherever it is.

And as we know, when your phones are turned on, they're constantly pinging your local cell towers, identifying themselves. In fact, that technology, we've talked in the past about, for example, how it's being used in shopping malls and things in order to, like, to anonymously, or maybe not as anonymously as we would like, to track people's habits as they walk through shopping malls - which windows do they stop in front of, how long do they stay there, how much do they use the bathrooms and how long do they stay there, all of that kind of information we now know is being collected for various marketing and demographic profiling purposes.

So the third thing that these guys did was they realized that cellular voice mail can be tricked. And there's something called "sly" something, slycall or slymail, I'm not sure, I don't remember now [slydial]. But it's a technique for sending two calls to a cellular number, slightly skewed in time so that the second call arriving finds the first call in progress and instantly goes to voicemail. It turns out that, if you skew these two calls, and you drop the first one, then you're able to get your second call to go to the cellular voicemail without ever ringing the handset.

And it turns out that at least T-Mobile's voicemail system is vulnerable to voicemail spoofing that allows this technology to access that destination phone's voicemail, allowing you to listen to all the messages that have been saved and get all the phone numbers of all the people who left messages.

Leo: Oh, man.

Steve: Basically allowing you - and these guys demonstrate this - allowing you to build a social network of, for example, you can determine using this all-legal hacking, none of this was illegal, no crypto was broken, this is just - although I wonder how people would feel about voicemail being slipped into. That seems a little wrong. I'm not sure that still qualifies as legal. But mostly they're leveraging technologies which are necessarily open, necessarily available, in order to determine, for example, where Brad Pitt is by name, then to confirm that it's probably the right Brad Pitt by slipping into his voicemail and seeing that he's got messages from Angelina Jolie, for example. It's like, okay...

Leo: This is him.

Steve: Nope, this is the right Brad. And essentially then get the phone numbers of those people and do the same thing to them to build a large graph, and to know where all these people are, more or less in real time.

Leo: Amazing.

Steve: So, yeah. It's a little bit of a wakeup call about what it means to have your cell phone on and be walking around. I mean, we're losing anonymity right and left.

Leo: It's kind of one of the principle techniques in that book "Daemon" and "FreedomTM," Daniel Suarez books, even. First thing people do when they stop, they, oh, god, they throw away their cell phone or they smash it or they break it because they know the daemon can track them.

Steve: Well, and it's a common theme now that we see in TV and movies is that your cell phone is giving away your location to the agencies that have access. And it turns out that you don't even need to be a government agency.

Leo: Anybody can do it.

Steve: You just need to be a couple of clever guys. Yeah.

Leo: Wow.

Steve: I wanted to mention that Mozilla has decided to blackmail, or blockmail, they call it - or I'm sorry, blocklist, not blacklist - blocklist the Java Development Kit which we talked about a couple weeks ago having a zero-day vulnerability. And you'll remember that Oracle, that now owns Sun, and thus Sun, was originally not going to be updating it because they didn't figure it was a big enough problem until it began getting exploited actively. And then they decided to push out an update, which they have done.

And so but in pursuing this, I ran across an interesting URL that I hadn't seen before which is the list of all add-ons which Mozilla is proactively blocklisting, which I thought was really nice. So it's www.mozilla.com/en-US. So in this case it's for English-U.S. So Mozilla.com/en-US/blocklist. And it's a not very long, but sort of interesting list of problems that the Mozilla team have found over time. And it's version based. So, for example, at the very bottom of the list, at the time of this recording, at the bottom of the list there is, sure enough, the Java Development Kit and the version number lower than which it will not allow Java to run in the add-on, to run in the browser. And I just want to give Mozilla and of course the Firefox team props for being this proactive. I mean, this is really...

Leo: No kidding, that's really great.

Steve: This is what we want. And unfortunately it's what we need. We need them watching out for us because not everyone is listening to Security Now! and making sure that their Java Development Kit is up to date as it gets changed.

In other news, it turns out that one of the patches which Microsoft released in the most recent Patch Tuesday, which was two weeks ago, April 13th, was a - some researchers have called it a "placebo patch."

Leo: Oh, great.

Steve: Because it does nothing.

Leo: Does nothing but makes you feel better.

Steve: It turns out it was to deal with a buffer overflow, a remote code exploit buffer overflow which was regarded as serious in the media services for Windows 2000 Server. So not something that's going to affect most listeners. Certainly it could be a concern for corporations that are still running Windows 2000 Server with publicly available media services being published. It's a buffer overflow in their Media Unicast Service. And Microsoft is going to be publishing - they've taken it out of the patch bundle now for any more patches that are being pushed. And they're revising it and are going to publish it next week.

Leo: So here's the question. Did they know it was a placebo patch? Or just what, I mean, what's the deal? How do you release a patch that does nothing?

Steve: It was just a mistake on their end. They believed that it fixed the problem. Somehow it came to their attention that after this patch was applied, the problem still existed. So they said, oh, whoops. And so they've stopped pushing it out, and they'll update it and push out another one. So I doubt that most listeners will even be aware of it. People who are running Windows 2000 Server will probably see it. Maybe it's granular enough that it will only disturb them if they've got media services installed. And Microsoft did their standard, oh, well, it's not installed by default. So it's like, okay, fine. So I guess that's good. And...

Leo: Broken by default.

Steve: Worth noting that we only have a few more months of Windows 2K Pro and 2K Server patches. That expires, the extended service period ends, on July 13th of 2010, of this year. So a few more months, and then no more updates for Windows 2000. It's really, really at the end of its life.

And then in one last little bit of news we talked about all of the brouhaha raised by CBS News's really interesting investigative report where they purchased a couple of copy machines that had the hard drives in them. One of them was previously leased by a company called Affinity Health Plan. And they have since, this Affinity Health Plan is a health insurance company, has acknowledged the data breach and sent out 409,262 notices.

Leo: Oh, boy.

Steve: So not quite half a million, 409,262, to all former and current employees, the providers they work with, job applicants, health insurance network members, and prior applicants for health insurance coverage, notifying all those people that their confidential personal data may have been leaked through the loss of an "unerased digital copier hard drive." So the good news is, this has made a large enough splash and caused a big enough problem that I hope that the news gets out, both to vendors of copy machines who really ought to provide some facility, not this, oh, for an extra $500 you can add the option which scrubs the drive. But, I mean, there ought to be some facility where, I mean, it just - somehow this thing reminds you that hard drives are going to be storing this information. Or just timeout the data. I do not understand how or why a copy machine has this wealth of information stored on it. But clearly it does.

Leo: There's a group, I don't know if you're aware of it, PrivacyRights.org, or the Privacy Rights Clearinghouse. And they do a running tally of how much personal information has been released in security breaches since January 2005. Now, this is only the stuff that's public, like the ones we just heard about. 353,812,819 records. That's pretty much everybody in the U.S. And these are the ones we know about in the last five years.

Steve: You have to be so far off the grid not to be caught in one of these.

Leo: It's amazing. I mean, I guess the question, and it's a legitimate question, is how much of this stuff that does get out is actually used against you, or is it just kind of in a dumpster somewhere? It's potentially a problem; right?

Steve: Yes. I would say potentially - one of the things that I've noticed, and I've noticed this really from sort of tracking the rate at which spam finds addresses...

Leo: A new address, yeah.

Steve: ...new email addresses, yes, is there's a really long lag. It's like this stuff gets out there, and it accretes for a while and sits somewhere. And then it gets purchased. But, I mean, it seems to take, like, most spammers quite a while, many, many, many months, to sort of finally say, okay, I guess we're going to start sending email out. Now, that, of course, that model may be very different for things like bank account logins. There you'd imagine that there would be much more notion of timeliness.

And we know in fact that some mischief is happening in near real-time. I did run across another story just during the week about the number of CAPTCHAs which are now being hacked by contractors in Third World countries who are well networked now, where they're only earning 10, $12 a day, but there are full sweatshops full of people sitting in front of computers that are doing CAPTCHAs. And as we know, that's got to be a real-time event. So somewhere someone, spammers, are being presented with a CAPTCHA because they want to create a spamming email account. And on the fly they send that into some network of CAPTCHA hacking where it pops up on a screen in China or India or somewhere, and someone there solves the CAPTCHA in real-time, sends the results back, and the spammer is able to convince the people who are protecting themselves that way that they are human and acquire a new account for spamming. So some things happen on the fly, and it's clear that other things take months.

Leo: And you know this, I don't think I'm revealing anything important, but you know this because you change your email address at the beginning of every new year.

Steve: Right.

Leo: So you know how long that new fresh address takes before it starts getting spam. And it's, what, you said it's a few months?

Steve: It's surprisingly long. And...

Leo: This is a good technique, by the way.

Steve: I do have Sue and Greg, we'll leave the support and sales email addresses active for several months into the next year so that anyone that we may have a dialogue ongoing with of course won't suddenly have it die on January 1st. And they will say to me, okay, can we kill last year's email address because...

Leo: It's all spam.

Steve: Yeah, there's junk coming in on it. So, yeah. But if you do that, it certainly is effective.

Leo: Steve does it algorithmically, so people like me know on January 1st exactly what his new address would be, so he can generate it.

Steve: Now, something I've been meaning to talk about for weeks that just sort of popped into my head I thought was extremely clever. And this occurred when I clicked the little Maps button on my iPad shortly after getting it. And it knew where I was.

Leo: Yeah, your WiFi iPad.

Steve: Yes. And...

Leo: Very accurately, by the way.

Steve: With startling accuracy.

Leo: Yeah, like within 80 meters or something.

Steve: It was amazing. And then I set about thinking, well, okay...

Leo: How do it know?

Steve: How does it know? And what's so cool, and maybe a little disturbing, is the way it knows. And that is that all WiFi networks, even when they are secured, when they are as encrypted as we could ever have our listeners encrypt themselves using state-of-the-art WPA2 encryption with uncrackable passwords, that encryption is carried within nonencrypted packets. That is to say that the payload of the wireless Ethernet packets which the WiFi system uses, that payload is encrypted, but the packet container isn't. And so...

Leo: What was that?

Steve: That's email coming in.

Leo: It's funny. Who is that? Bamm-Bamm? What is that?

Steve: It's just a WAV file.

Leo: You've got mail.

Steve: Yeah. I found it on CompuServe years ago.

Leo: I love it. They found your address, Steve.

Steve: Someone's just sent me something. So what I'm saying is that, even in an encrypted network the MAC address of the access point is known and is visible to everyone. And MAC addresses are unique. You can manually force them to be something. But from the manufacturer they're a 48-bit address, 24 bits being a manufacturing ID, the other 24 bits being a unique serial number within that manufacturing ID. So what has been done, and I guess this was Google rolling around...

Leo: No, no, no, it's a company called Skyhook.

Steve: Oh, Skyhook. Now that you say it, I remember that.

Leo: I mean, there are other companies probably, but Skyhook's the big one.

Steve: Rolling around our city streets and residential neighborhoods, they had antennas out, and they were acquiring the MAC addresses of all the access points that were available.

Leo: If you think about it, it's kind of clever.

Steve: It's very clever. I mean, it's really, I mean, I would say it's really cool, if it didn't sort of give me a little bit of the willies from a privacy standpoint.

Leo: But they're not using anything that you aren't broadcasting to the world.

Steve: That's true.

Leo: All they have is a GPS in their truck and basically Stumbler. And they say MAC address, GPS; MAC address, GPS. And they triangulate.

Steve: Yes. Well, and, see, that's just it, is they've got - they have signal strength and all the MAC addresses. And I'm sure people who have laptops will have the experience I have of pretty much wherever you are these days, if you look at, like look at all the available networks, you'll see maybe a list, I mean, from my own location, there's, like, 12 that are, like, around me. And so most of them, I'm happy to say, most of them are encrypted. But that doesn't matter because MAC addresses are not. The contents of the packages are. The MAC addresses are not, which I thought - which is the little moment of aha that I had when I realized that's how this is being done. And so if you've got WiFi and access to this database or one of these databases, it's pretty much possible to know where you are all the time.

Leo: If I hide my SSID, does that break it?

Steve: Nope, [indiscernible].

Leo: No, because there's still packets going out.

Steve: The only thing you could do, if you really wanted to, like, obscure this - and, I mean, you have to do it deliberately - you could go, like, get the MAC address of an access point, like somewhere else in the country.

Leo: Spoof your MAC address.

Steve: Yes, and manually change your MAC address. Now, that would not cause a problem because there's no chance of a WiFi collision. That is, the MAC address being broadcast here and on an access point in New York - I'm in California, someone else with the same MAC address is in New York. Well, the only problem that you'd ever have would be if those two access points were within radio range of each other, which because they're at opposite ends of the continent would never be a problem. In which case, that would really foul up anyone who was using this technology to locate themselves because suddenly they would think they were in New York if they happened to be close to mine. But you'd imagine, too, that this technology, if it was good enough, it would say, wait a minute, we've got one outlier and three others do look like they're valid. So discard this one that seems to be some sort of a mistake.

Leo: If you're in an urban area, or at least an area where there's some population density, this always seems to work. It doesn't work well - of course if you're not in your WiFi, it doesn't work at all.

Steve: Right.

Leo: And it may not work well if you only have one WiFi access point nearby. But anywhere densely populated it's pretty - you're right, I noticed it with the iPad. I've known about it for a long time. But it's amazing. You don't really need GPS.

Steve: Yes, I mean, it nailed me just absolutely. And I looked at it, I thought, okay, I'm going to do a little research here and figure out how this happens.

Leo: Interesting.

Steve: And what I loved was, and I just wanted to share with our listeners, was the realization that access points are generally not moving, they're in a fixed location, and their MAC addresses are completely available, being broadcast whether they're encrypted or not. So that creates a wealth of reception-point opportunities. And you could imagine that this - I want to call them Skynet. You said it was Skyhook.

Leo: Skyhook, but it might as well be Skynet.

Steve: Might as well be Skynet. They've got super-sensitive receivers on their trucks, more sensitive than we normally would use or need, because it's allowing them to do ranging by virtue of signal strength. And just it's very clever. I thought it was neat.

Leo: In the chatroom somebody's saying that Google also does this with - and it would make sense. If you're driving around, you might as well.

Steve: Yeah, why not. If you're taking pictures of sidewalks and all that, might as well suck in as much information as you can.

Leo: Why not?

Steve: Yeah.

Leo: Tie it all together. The more we know...

Steve: Archives are big.

Leo: Yeah, yeah. Well, it's really changed - that's what I think the big change in privacy is, not so much that we expose ourselves more, that's part of it, but not really the big part of it. The big part of it is that computers are really adept at taking disparate databases and cross-referencing and drawing a picture of somebody. So when all, I mean, look, real estate records have been published publicly in courthouses for centuries. It was only when they went online, when somebody went to the courthouse, typed them in, and put them online that it became a problem.

Steve: In fact, a buddy of mine recently purchased a house, and he said you wouldn't believe the amount of official-looking junk mail that I'm receiving because somewhere there is a record that - I'm sure someone's selling the database of new home purchases. And marketers are purchasing them and sending them, oh, look, you owe money.

Leo: In many cases these local governments are not putting this data online. But people go to the courthouse, get the public records on paper, and type them in.

Steve: There's a market for it.

Leo: There's a heck of a market for it. And just use Zillow sometime, Zillow.com, you could find out what all your neighbors are making, what their houses are worth, all of that. It's kind of scary. But it's all public information. It's the power of a computer to collate this information, cross-reference it, and make it available that really has transformed everything.

Steve: Yup, aggregation.

Leo: Aggregation. We have some great questions from our audience. As always, Steve has collated the best, most representative questions, 11 questions good and true.

Steve: And a large dose of, not surprisingly, iPad security issues. I asked for them, and we got them.

Leo: Oh, good.

Steve: So we're going to beat this thing to death today. And I want all of our listeners to know, at that point we're done.

Leo: Oh, it's worthwhile. Look, they're selling these at a rapid clip.

Steve: More than a million now, Leo.

Leo: More than a million now. And I think they may become a very ubiquitous computing platform. So it's really appropriate to address security, particularly since you cannot put any security software on this thing.

Steve: Right. And there's a question about that, as a matter of fact, this week.

Leo: Yeah. What you see is what you get, so how good is what you get? All right, Steve. I've got questions for you. Are you ready?

Steve: You betcha.

Leo: Starting with Greg Christopher in Silicon Valley. And again, this has to do with security. He says Apple's restrictive development tools SDK agreement doesn't do anything for security. Hi, Steve. Thanks as always for another great show. I listened with great interest to your discussion around the iPhone security model. The iPhone is still using a lot of OS X underneath, so there is a lot of potential for entry points by hackers.

We know very well - this is parenthetical, I'm saying this, Marc Maiffret has pointed this out - that there's a lot of holes in OS X, especially in this open source software. I am in agreement that the App store is a good defense against spyware and against viruses creeping into the iPhone the way they have on the Android platform. But I did disagree with something you said. When discussing the SDK agreement, you brought up the thought that the new wording in the SDK agreement actually increases security. It does not. Apple said in this new SDK you may not use third-party tools, you can only use Apple tools to develop for the iPhone and the iPad.

He says: The problem is that Apple is now dictating how you develop to their APIs. The technology recently developed by Adobe simply takes a Flash application and reconstitutes it to be a compiled program that calls the Apple APIs to do its work. There is no system-level Flash interpreter, nor native code that circumvents those APIs. It is all going through standard documented Objective C APIs. Apple's agreement is simply there to make it harder for developers to make something that works on the iPhone and, let's say, Android, or Palm, or Windows Mobile.

In essence, when Adobe changed their application development

environment to conform with Apple's rules, Apple changed the rules. And they did so about four days before a multi-year development effort was supposed to be released to customers. It's hard to look at this in any way but as anticompetitive. Here is the wording from the Apple SDK agreement:

3.3.1 Applications may only use Documented APIs in the manner prescribed by Apple and must not use or call any private APIs. Applications must be originally written in Objective-C, C, C++ or JavaScript as executed by the iPhone OS WebKit engine; and only code written in C, C++ and Objective-C may compile and directly link against Documented APIs, (e.g., Applications that link to Documented APIs through an intermediary translation or compatibility layer or tool are prohibited).

While Apple could say that user experience might suffer due to lack of the normal interface, there are no human interface guidelines for the iPhone, and every application behaves differently since most buttons are nonstandard and menus nonexistent. So from a security perspective, I don't think we've gotten anything here. But from both an iPhone user and iPhone developer perspective, we're being given less choice.

Thanks again for the great podcast, which I highly value for its technical, newsworthiness, and entertainment value. Greg Christopher. Steve?

Steve: I agree.

Leo: Simple as that. Nothing more to be said here.

Steve: I don't remember...

Leo: I don't think we said it was a security thing.

Steve: Yes. I don't remember saying that Apple changing that enhanced the iPhone OS security. But I wanted to correct the record, if that was the impression that we gave. And I do agree that I think this particular decision, it's hard to see it as anything but what really does appear to be an Apple vs. Adobe war going on. Is that, I mean, is that...

Leo: There's one argument, and when somebody emailed Steve Jobs and said what's the story, Steve pointed to a John Gruber blog post. He said, "This explains it." And you're absolutely right. There's huge anti-competitive elements to this. But the one argument you could make in its favor is that, if you allow a third-party tool that develops write once, run many, like Java, to a lot of different platforms, then users of those programs no longer get to use - Apple no longer drives the features available to them. This third-party interpreter does. Because only features available to the third-party interpreter are available to the user.

Steve: Lowest common denominator now.

Leo: Lowest common denominator. Now, you might say, well, yeah, okay, but we'll make that deal with the devil to have all this new development. Apple's basically saying we don't want some other third party to drive what features are available on the iPhone. If this tool suddenly becomes very successful, it's out of our hands now. We can add a feature to the iPhone that's not available to users because they're using these applications developed by this third-party tool. So I think that that's a factor.

Steve: I was going to say that sounds a little bit like, okay, how can we justify what we want to do?

Leo: Yeah. I mean, clearly there's economic benefit to Apple doing it that way. I think you might say there's some security benefit because of course an interpreter could always introduce flaws. I guess he's saying it compiles down to native code, so by the time the user gets it it's not using the interpreter. Is it possible, can you see a scenario where that still could add a security flaw? I guess if the libraries and the APIs are all secure, let's say they're a hundred percent secure, using an interpreter that compiles down to API calls, that would have to be secure.

Steve: I think it would be more secure, actually. To use something which is going to create a layer of automation between you and the way you use the API would tend to prevent you from making a mistake with the API.

Leo: Right. Unless Adobe made the mistake.

Steve: And they would never do that.

Leo: Mistakes would be centralized.

Steve: Adobe never makes - remember, Leo?

Leo: But mistakes would be centralized and because of that would be easier to fix.

Steve: Yes, yes, yes. And probably, one would hope, it would get caught early on. So...

Leo: We've seen, I mean, Microsoft's had problems with - remember the exploit with metafiles, with its metafile libraries. Everybody used that library, and as a result had problems. But that's not what we're talking about here because it wouldn't be a library that would be used by the code. It would only be the library used by the code to compile to the native API.

Steve: Correct. I really do agree with Greg and the point he raises. I think, to me, this really seems arbitrary and mean-spirited. And the flipside is I don't think we're going to be at any - we're going to have any lack of applications for the iPad. Where are we, at like 180,000 for the iPhone now? But I don't disagree. So if I ever gave, if I did give the impression that this restriction was a security benefit, then it's hard to find one.

Leo: Right. Next question from Corry Macfarlane in Minneapolis. A fix for the KatMouse problem with Firefox.

Hi, Steve. First, I've been listening since day one. Love the podcast and have turned many others on to it, as well. I'm also a user of SpinRite and have been for quite a while. It's been a life saver many times over, really.

You guys may have covered the solution already but I don't recall hearing this exact solution, and it has worked every time for me. I just upgraded to Firefox 3.6.3, and of course KatMouse quit working under the new Firefox. KatMouse is a great program Steve has recommended several times that changes the scrolling. Is that right?

Steve: Yeah. What it does is - everyone I've recommended it to is unable to live without it after they use it because it - normally in Windows you need to click on something that is scrollable, like your web browser or Notepad or Word or whatever. Anything that scrolls, you have to give it "focus," as the programmers refer to it, and then your scroll wheel will scroll it. KatMouse is a little smarter. It looks at what you're floating over, whether it's got focus or not, and it sends the scroll messages to that window, even if it's not the topmost window or doesn't have focus. So once you get used to it, and if you have a mouse with a frictionless scroll wheel, it's just, oh, it's just wonderful. But I talked about how, and my tech support guy Greg had the experience, of when we upgraded to Firefox 3.6.3 it stopped working.

Leo: So here's the fix. The original post can be found on the Mozilla Support forums since it has caused so many KatMouse users grief. So Mozilla's published a fix. It's Item 571918, if you want to go to support.mozilla.com, 571918, about halfway down the page. KatMouse works with Firefox 3.6. You just have to configure it. Right-click on the KatMouse taskbar icon. Choose Settings. Select the Classes tab. Drag the target icon at the bottom of the KatMouse window onto any Firefox window and release the target. "MozillaWindowClass" will appear as a custom configuration. You can double-click on it for further configuration. You don't need to in this case. Apply it, and you're done. You're teaching KatMouse about Mozilla's windows.

Steve: Exactly. And essentially what you're doing is you're saying this window already knows how to handle scroll wheel messages, windows messages about the scroll wheel. So it's not necessary for KatMouse to convert them into up and down arrow messages for the window. And I wanted to acknowledge all the listeners. I've been intending to do this in a Q&A for many weeks. And I just, again, I ended up finding so many other good Q&A questions that I said, okay, well, I'll just - I wanted to mention this in errata, but I kept forgetting. So now we're on the record.

Leo: Now you know.

Steve: It is absolutely possible for KatMouse to be configured to be completely happy with Mozilla Firefox.

Leo: By the way, this applies a little bit to our last question because the reason that didn't work in Mozilla is because Mozilla's not using a standard windows class, it's using its own windows class, not a Microsoft windows class. And that's the kind of thing that happens when you allow developers to create their own - the more open the platform, the more people are going to do that. And so KatMouse makes an assumption, well, I know windows, I know Microsoft windows classes. I'm going to be able to handle those. And if you do something nonstandard, it's going to get confused.

Steve: Right. And it may well be it may hail from Mozilla's a cross-platform...

Leo: That's what I think. It's a cross-platform thing. And so that's why I say that's the same thing.

Steve: Exactly. Very good point.

Leo: Yeah. When you develop cross-platform, sometimes you break stuff in the native platform.

Steve: Right.

Leo: Tony, listening in Yokohama, Japan has some comments about iPhone OS 4 and its security improvements: Steve and Leo, I was just listening to your most recent Q&A, and I thought I'd bring up an interesting fact about the iPhone OS. One thing that has always bothered me about the iPhone locking feature is that it's a simple four-digit pin. It's on the iPad too that way. You've mentioned before about complex passwords and password length, so I'm sure you can sympathize.

Well, in the new 4.0 OS, or at least the beta, you now have the option of using a complex password to lock the phone. This enables the full keyboard, special characters and all, and still supports the 10-strikes option. That's the one I talked about the other day where, if you try and fail 10 times, it erases the data. Unfortunately, the beta had quite a few bugs, so I'm back to OS 3.1.3. But I'm anxiously awaiting the final release, or at least a more stable beta. They don't have iPads where I'm at yet, but I thoroughly intend to get one as soon as they do. Having access to a full password to lock the iPad would be a great feature, and I hope 4.0 is released for both platforms at the same time. Actually Apple has said that the phone will come out first, iPad will be in the fall. Probably about the time Tony gets his iPad in Japan. Maybe even sooner.

Thank you for the great show. Keep up the great work. P.S.: I'm a registered iPhone developer and was using the legitimate beta from Apple's website. No jailbreaking for me. Good.

Steve: Well, I wanted to remind our listeners about that. My iPad got that lock put on it, like within seconds of us ending our recording last week, when you mentioned it. It was just one of those, duh, why am I not doing that? And, you know, it's very breezy to type in a four-digit code. I also activated the wipe memory if I miss it wrong 10 times. Can you set that, or is it always 10? I think it's always 10.

Leo: It's always 10.

Steve: Because I'd turn my down. I mean, I don't think I've ever mistyped it, but I might set it to three.

Leo: Oh, you will. Three might be a little fewer than you want.

Steve: I think that under this OS, what, we're at 3.2 right now? I believe that the corporate configuration pack allows even this one to accept longer or full keyboard settings. I ran across some reference to that when I was reading through the developer docs two weeks ago. But for me, four digits is fine, given that 10 strikes and, boy, are you out.

Leo: That's, I mean, that's the key is, I mean, four digits you'd have to guess, what is it, an average of maybe 500 times, something like that? What is the average? If you have a thousand possible choices.

Steve: Right. A thousand possible choices.

Leo: Actually you have 9,999. You have 10,000 possible choices. What is - so what would the average number of times, what would be the predictable number of times you'd have to try? Half?

Steve: Yes, it would take - on average, guessing at random, assuming that there was no pattern that was like 0000, yeah, you would expect that it would take you - that if you guessed 500 you'd have a 50 percent chance of hitting it. And certainly long before then the thing will have wiped itself out, so.

Leo: Unless a person's lucky. Now, somebody, I think it was Dvorak...

Steve: You could get lucky?

Leo: You can get lucky. Dvorak - I mean, you'd have to get pretty lucky.

Steve: Yeah.

Leo: Dvorak - it's one in 10,000 each time; right? Dvorak said, now, don't forget, you can look at the fingerprints and maybe - maybe something. And he tried to hack my iPhone looking at where the fingerprints were.

Steve: And that's interesting that he mentions that. There's one, I found a couple of really wonderful puzzles. And one of them, you're inherently dragging your finger in a grid pattern as you sort of, like, drag these colored strips around to, like, weave them through something. And so it was the strangest thing because, like, the next morning, before I turned the iPad on, it was still dark, and I have an antiglare covering, top sheet on mine, and the entire thing was, like, had this, like grease in a grid, my finger grease just in this perfect grid because I'd been - I'd spent an hour dragging it around in these little street grids. I thought, yeah, it's like, whoa, you really can see where you've been. So I see John's point.

Leo: Yeah. I'm glad he didn't try 10 times, or I would have lost all the data on my iPhone.

Steve: Yeah.

Leo: Thanks, John. He is trouble, I'm telling you. He comes in here...

Steve: Actually that's a very good point. I mean, not that losing all the data on the iPad is a bad thing necessarily because...

Leo: You synch it.

Steve: ...it's being docked and synched and so forth. But you could maliciously wipe out someone's iPad just by deliberately being wrong 10 times in a row.

Leo: I had to take it away from him before he did that.

Steve: Yikes, yeah. Bad John.

Leo: He did that with Mint, though, because I have Mint, and he was trying to get my financial records. And he was - and it also has a four-digit code on it. And he was trying and trying and trying. And eventually Mint said, no, you've got to log in again now, dude.

Steve: And, by the way, we've talked about PayPal allowing you not to use your little security dongle. What I learned was that it only allows you not to use it maybe two or three times. And then it finally says, okay, this is your last login without using your dongle.

Leo: Oh, good. Well, there you go.

Steve: And I thought, well, that's good. So I wanted to mention that to our listeners.

Leo: Better than nothing, anyway.

Steve: Yeah. Well, better than just allowing you not to ever use it again.

Leo: Right. You don't need that dongle that we gave you that you spent all that energy setting up. What do you need that for?

Steve: Yup.

Leo: Question 4, Jim McShaver in Saskatchewan noted something disturbing: Steve and Leo, love the show. In regards to the iPad being the most secure device for banking, I don't bank over WiFi anymore. The iPhone, and I would assume iPad, remember only the SSIDs that you have trusted. Okay. Unfortunately, they don't tie the SSID to a MAC address, even though that is spoofable. I own three wireless access points and have tested this. So if you have ever connected to "Linksys," "D-Link," or "Steve's Starbucks," it will connect automatically to any other access point with the same name. This is also true of PCs, but they are not always with us. That's not right. That's bad; isn't it?

Steve: Yeah, it is. But it highlights something that I think is important. So just to summarize, what Jim is saying is that it's the SSID, the so-called "beacon" which the access point broadcasts announcing its identity, that's what our WiFi clients are matching on when they say "reconnect without asking in the future." So access points which are left set to their default of Linksys or D-Link or whatever, actually Steve's Starbucks is actually just AT&T WiFi. And I'm sure that every Starbucks has the same SSID, which is probably, you know, AT&T WiFi.

So what he's saying is that, while it would be possible to lock to the MAC address, where you'd be then, as we were just talking about in this geolocating example, you'd be locking to a unique access point rather than locking to the substantially less unique SSID name that the access point was given. But it's almost the case, I think, that doing that, I mean, yes, I would prefer that it was more discriminating, and it would be nice if there was an option and you could say use the name or use the uniqueness of the access point. You'd have to dumb it down for most users because they don't know what a MAC address means.

But Jim is saying he doesn't use - he doesn't do banking over WiFi, which I can certainly understand. But in all cases we're typically talking about open WiFi, that is, when I'm using the iPad here at home, I've got deeply encrypted, you know, the best encryption we can, with impossible-to-manually-enter passwords from GRC's Perfect Passwords page. So I'm very secure there. But when we're out roaming, we're often in open WiFi hotspots. So it's crucial that nobody trust those hotspots. That is, the fact that the iPad is associating with the access point and connecting is, whether it does it with you saying yes I give you permission or not, to me doesn't really matter a lot because you've got no security on an open WiFi connection anyway. And you should never, under any circumstances, consider doing anything that requires encryption.

So, yeah, it'd be nice if it asked you for permission, and you could configure the iPad and the iPhone so that it will not automatically reassociate without your permission, which anyone with this concern should do. But I want to make sure that everyone understands the inherent danger in open WiFi settings. We had a Q&A a few weeks ago where someone reminded us of this because a friend of his saw somebody running WiFi sniffing software on a laptop at a facility, at a location where there was open WiFi. The person was clearly collecting usernames and passwords and email and who knows what from everyone who was there. We just - that danger has to be foremost in people's minds.

Leo: If you're SSL to your bank or to your email on an open access point, you're okay, though; right?

Steve: Well, I wish it were true. But we've seen how weakened SSL has become.

Leo: Right. There could be a man in the middle.

Steve: I mean, it's just, yeah. You want to make sure that you're authenticated, you're connected to the organization that you think you are, using all the tools that are available.

Leo: Using WPA2, if possible.

Steve: Yes, WPA2 to encrypt locally. And look at the security certificate and see that it makes sense. Unfortunately, I don't think you can even do that with the iPads. And there's a question that sort of bears on...

Leo: Oh, that's an interesting point, yeah.

Steve: Yeah, there's a - toward the end of this list of security concerns there's a question that bears on that, though, that we will get to shortly.

Leo: Here's a kind of bizarre little aside on this. If you did use WPA2, even if the SSID was the same, obviously if the passwords were different it wouldn't auto-authenticate.

Steve: Correct.

Leo: What if the passwords were the same? Which is probably a highly unlikely scenario. Maybe Starbucks, though, let's say Starbucks started to use WPA2. Or I've got a better idea, you have a coffee shop that you go to, and they have three stores, and for convenience they use the same SSID and password on all three stores so it's automatic.

Steve: Yeah, and it would connect.

Leo: It would.

Steve: Without caring, yes.

Leo: Okay. So it's not doing anything else.

Steve: Although you'd also have security in all three cases, so that would be good, as long as the password was useful. And, see, that's another problem is that, for example, there's an Italian restaurant that I like that has a big "Free WiFi" sign on the front door. And the first time I went in with my iPad, I said, hey, I've never had an occasion to want to be on your WiFi network, but now I have that occasion because I have an iPad. And it was, I don't know what it was like, it was, well, whatever the password was, the waiter just gave it to me. So it wasn't open, and it was encrypted, but any customer asking could get the password. Of course we know what that means. That means that, even though you are in a secure network, everyone there has the password, meaning that anyone can listen in...

Leo: You're still on a public network, in effect.

Steve: Exactly. You're back to the exact equivalent of open WiFi because the password, even without it being complex, it's something that anyone can know, and that's all it takes then to be able to decrypt everyone's traffic.

Leo: Hysterical.

Steve: Yeah.

Leo: I love it. You know what I really enjoy is how clever and thinking our audience is. They're always thinking about this stuff.

Steve: Yes, yes.

Leo: Vicissitudelicious in San Jose - that is the longest handle I've ever heard of, and most difficult to spell, by the way. Vicissitudelicious in San Jose asks about ShieldsUP! and stealth ports: I have been trying to stealth my ports and have been unable to find a way to stealth them. I am using Windows Firewall. It seems that one can only turn ports on, but not off. I also have tried using Comodo free firewall, same problem. When running ShieldsUP! my ports are all closed, but none are stealthed. I used to have three ports stealthed, but not anymore.

I noticed that a number of people on the Internet have had the same problem. How do you stealth a port? The only time I had all ports stealthed was many years ago when I had ZoneAlarm, which seemed to automatically stealth all ports. And that's because of Steve, by the way. In fact, Steve's the guy who invented the whole thing about stealthing ports. So you're asking the right guy.

Steve: I coined that term, too.

Leo: Yeah. Will you ever mention how to stealth ports on a future show? Or maybe I missed it in a past show. I have listened to almost of all your excellent shows, but I'm getting old. So please forgive me for not remembering if you've covered this already. We cover it periodically, but it's a good thing to re-cover, I guess. I used to work on Federal Systems mainframe communications computers that had plated wire memory. That's core; right?

Steve: It's sort of a predecessor to core.

Leo: Pre-core [whistling]. 32k was huge. Well, yeah, if you've got to wire it all up by hand, 32,000 connections. Or more than that; right? Eight times 32,000. Also, I had to enter hexadecimal instructions using rocker switches. Steve had to do that, too. So we've really come up in the world. I've started to use Vitamin D. Thanks for the advice. I'm hoping that Vitamin D is the firewall that will prevent the biological viruses from getting in. I hear zinc is good for that, as well. So, stealth. Let's talk about stealth.

Steve: Yeah. It occurs to me the reason that three of his ports may have once been stealthed was that at some point an ISP was blocking, for example, the Windows filesharing ports, which, for example, my own cable modem provider, Cox, also does.

Mostly I'm concerned that he doesn't have any stealthing because it sort of implies he's not behind a router. And he's clearly concerned about security. And routers are so inexpensive these days. I mean, they're like sub-$50, and trivial to add to a network. The idea of not being behind a router just sort of gives me the willies.

So, I mean, and to the best of my knowledge all routers are stealthing now because it's just become the thing to do. I mean, again, I have to say I think it's ShieldsUP! is largely responsible for that behavior because earlier routers weren't, and people were complaining, and routers changed their behavior so that people would stop complaining. There's been controversy, sort of constant controversy whether, especially among the old UNIX curmudgeons, about whether this whole stealthing thing is worth anything anyway. It's technically a breach of the Internet RFC rules to have any machine on the 'Net not respond to a ping. They're all supposed to. And ports are supposed to answer that they're closed, rather than doing nothing, which is what they do when they're stealthed. I've always said, yeah, well, okay, I accept that. But isn't it nicer to appear not to exist at all than to be an obvious machine on the Internet that then in some way encourages people to poke at you more. So that's been my argument for it.

Anyway, to Vicissitudelicious, I would say, wow, get a router, and stick it between whatever connects you to the Internet and your machine, and you will be stealth. But more importantly, you'll have a layer of hardware firewall security which you lack right now. It's too easy for something bad to get into your machine and turn off the Windows Firewall or to open ports through it by using the Universal Plug and Play technology that Microsoft has heralded. And so it's just so inexpensive and so easy to do. And then you get the ability to share your Internet connection with more machines. You get the benefits of a router. But also just the great security, which is inexpensive, of having something outside of your Windows machine that's protecting you from the 'Net. The 'Net's just - the idea of plugging a Windows machine directly onto the Internet [vocal shuddering]. Yeah.

Leo: Gave us chills. And not in a good way. Question 6, Completely Anonymous, somewhere on the 'Net, writes TNO but Apple. TNO is Steve's acronym, Trust No One. But Apple? Steve and Leo, towards the end of last week's episode 245 you started trusting "closed source" vendors, whereas early in the podcast you were all gaga about TNO. So the conclusion sounds like TNO except Apple, Microsoft, et cetera. Is that what you meant to say, Steve?

Steve: Well, now, it brings up a really good point because the only way you can truly trust no one is if you go out onto the beach and get a bucket of sand, and set up a semiconductor fabrication facility in your garage, and make your own chips which you design, and hope you don't make any of your own mistakes, which are a lot more likely probably than Intel making a mistake. And then you sit down and write an operating system and all the required utilities and application software and so forth. And basically, from the silicon up, you build a machine. Short of that, you're trusting someone.

Leo: Right.

Steve: So we understand that there are always tradeoffs. Security is not perfect. Companies are not perfect. What you want is you want to choose who you're going to trust, and you want to trust as few people as possible, or trust the most trustworthy ones and have the wisdom not to trust the ones that you shouldn't trust. This is all gray. I mean, I wish it were black and white. There isn't any black and white. So do we trust the people that build our laptops not to put bad stuff in there?

There was a horrible and amazingly persistent rumor on many websites, I'm sure you ran across it, Leo, where people were sending pictures of a keyboard recorder that was supposedly embedded in laptops. And it wasn't. It was just an urban legend that, because the photos looked convincing and the write-ups looking convincing, everyone kept passing it around and worrying that the laptops had built-in keystroke recorders. And they don't. But we assume they don't. I mean, we know they don't. But they could. And but there's just - it's just vanishingly unlikely that Toshiba or Sony or HP are going to build keystroke recorders into their laptops because it would just be the end of the company...

Leo: Right.

Steve: ...if they were to do so. And there isn't any reason for them to do that. Similarly, do I trust Apple? Well, I trust Apple's intentions. I don't really believe Apple has an economic motivation to maliciously do something. Neither do I think Microsoft has that. There was that NSA nonsense in Windows for a while, some DLL or something in the kernel talked about NSA, and everyone thought, oh, that meant that the NSA had backdoors installed in Windows. And it's like, no, that's not what it means. So...

Leo: The risk, the commercial risk to their reputation, it just way outweighs any other benefit they could gain by doing that.

Steve: Yeah. And there are ample opportunities for bad people to get software in our machines. To me, that's the risk. Now, when I talk about Trust No One, and I use that acronym, I mean, I think it's a nice reminder. And for me it says, if there's an architecture which requires that I trust versus does not require that I trust, and I can use either, I'm more comfortable with an architecture that does not require that I trust.

Leo: It's just harder to say all that.

Steve: Yeah, exactly.

[Talking simultaneously]

Leo: The acronym's longer.

Steve: TNO works.

Leo: Yeah. Trust [laughing] no one, well, you have to, but trust - never mind. Bill Newhouse, Rockville, Maryland asks about a single search to rule them all. Guys, is there a way to search into all episodes of Security Now! via a single search? For instance, I might wish to know in which shows you discussed DNS. Searching show by show is painfully slow. I just discovered the TWIT wiki and recognize that wikis are good for such searches. You might wish to highlight this notion in a future episode. Thanks for the informative, useful, and thought-provoking show. Of course the wiki isn't perfect because we haven't been updating it since the beginning. It's relatively new.

Steve: I wanted Bill Newhouse and all other listeners to know that GRC has all of the transcripts being searched and indexed by Google. And if you go to GRC.com/sn, thanks to listeners on the show and Leo liking short URLs, GRC.com/sn will take you to the Security Now! page. And there is a search box, as there is everywhere on GRC, in the upper right-hand corner, up in our site menu. And you just put whatever you want to in there. It will - and that's a search by Google that's been customized.

You do have to have JavaScript enabled. And I've been intending to look into that because a couple of people have said, hey, your search doesn't work, and I'm pleased that they had JavaScript disabled. But I just used the code from Google that not surprisingly requires JavaScript since Google is JavaScript land. But if you do have JavaScript enabled, the search will work, and it will find every - because Elaine went back and proactively did transcripts on all of the earlier episodes after we started using her to create weekly transcripts, everything is indexed, and you'll find every reference to DNS or anything else we've talked about. And so it's really cool. Leo was talking at the beginning of the show about here we are approaching five years of archived content. And it's all indexed and searchable.

Leo: I should give credit to you because Elaine didn't do it just on her own recognizance. You paid her to do it. So Steve has, out of his own pocket, paid for all of these transcriptions, and paid for it going back in time to do that. So thank you, Steve.

Steve: Yeah, GRC and Security Now! are a substantial portion of Elaine's income, so...

Leo: Yeah, we keep - you keep her busy.

Steve: I'm glad to do that.

Leo: Yeah. Well, thank you. I should probably help you with that. We should talk. I feel bad now. Maybe I can write you a check. Number 8, we've got three more. DuckByte in Mission Viejo, California asks about DHCP vs. Static IPs. I've been wondering this myself. I'm really glad he's asking. Steve and Leo, I have about 10 different devices connected to my home network. Automatic assignment of internal IP addresses is normally not a problem. But one of the devices is used as an FTP server; and, when the internal IP address changes, I have to change the router and server settings. I guess he's doing port forwarding or something. Is there a way to configure the network so the mix of static and dynamic IP addresses can be used? And in face I'd like to add to that. When should you use static? Is it okay to use static? Is DHCP okay? What do you do? You probably use static. You seem like the type.

Steve: Yeah. My entire network, I've just never even had DHCP in here. Although one of my wireless routers is, so that it just - laptops don't have to be configured. But another one is set up as an old-style access point rather than as a router. So here's the deal. He's talking about not external access to his FTP server, that is, he's not trying to get to it from outside the Internet. He's just saying within his own home local environment he would like the IPs of at least that one machine, that one machine that is an FTP server, he wishes that its IP on his own network, 192.168.0.whatever it happens to be, wasn't whatever it happens to be.

There's two ways you can achieve that. Most routers themselves now allow you to associate a MAC address with an IP. And that would - that is to say, the MAC address of any of the devices on the network. So, and this is specifically to solve this problem. So using the router's user interface, you can - there's normally a way, there's an option on the menu where you can show all the clients that are currently connected, that is to say, all the devices which have obtained a DHCP lease. And we've talked about leasing because, in fact, just last week when we were talking about the iPad's little glitch with its not releasing its lease if the screen had gone to sleep on you.

So what that list normally shows is the MAC addresses and the current IPs that had been assigned by the router's DHCP. So what that allows you to do is determine the MAC address as the router sees it. So you would look at, for example, what the IP is of this FTP server now, and then you look in the table, in the router, and you'd find that IP and the MAC address of that machine. Then in a different area of configuration in the router you're normally able to say, I want to assign static IPs one for one to MAC addresses. And so you could choose, for example, 192.168.0.20, just to sort of have it, give it a nice number, and kind of keep it out of the 12345 range that all of your floating dynamic IPs are going to have.

And the way you do it is you say to the router, give this IP always to this MAC address. So every time, any time that machine turns on - the MAC address, remember, is actually the fundamental way that devices on an Ethernet LAN identify themselves. So that machine will be broadcasting, saying hey, I need my IP configured for me. Give me an IP. The router receives that broadcast and says, ah, this happens to be a special query because it sees that it's coming from this particular MAC address. So it always gives it the IP that you've assigned it, 192.168.0.20, for example. So that's one way to do it.

The alternative is, if you didn't have a router with those features, or if you want a router-independent solution, you can simply tell that machine that has the FTP server to not obtain its IP automatically. That is to say, in Windows language we're used to seeing our interfaces set to obtain IP address automatically. If you just change that to something that is to assign it manually, then nothing prevents you from setting it yourself, that is, on that machine that has the FTP server in it, or any machine you want to have be a fixed IP, set it to something like 192.168.0.20. Now...

Leo: You just have to make sure to avoid collisions, that's all.

Steve: Correct. I was just going to say you would - there's two things you can do. Normally the range of IPs that will be issued by DHCP is settable in your router. You might say, in the configurations, it'll normally have like a lowest and a highest, a starting and an ending number. So, for example, 192.168.0.1 through 192.168.0.50. Now, it's normally the case that even though it's only going to assign IPs in that range, 192.168.0.anything would be valid. So you could simply set the IP of this machine to .100, that is to say, outside of the 1 to 50 range. And so it's got a nice number, .100.

When you turn it on, since you've set it to have a static IP, it doesn't ask the router for an IP. It already knows what its IP is. It's .100 always. And then all of the machines that are getting floating random IPs, they're always able, no matter what their IP is, to access the FTP server at 192.168.0.100. And either approach will work.

Leo: So I want some guidance from you, Steve. We're running out of time. We have about five minutes before we have to do "This Week in Google." But we have a bunch more questions. So, and by the way, the remaining three are all about iPad security. Do you want to pick one and do it? Or do you want to save these for another date? What would you like to do?

Steve: Let's just pick one, and we'll - I don't really have a preferred one. So let's just do #9, and we'll do 10 and 11 next time.

Leo: We'll save them. Alec Stubbs in the U.K. has a thought about iPad security: Steve, I've been a listener since day one. I really enjoy it. I have a quick response to something you said in the last show. You made the point that competing ecosystems to Apples iPhone OS lose out somehow because they don't test and sign applications that are allowed to run on their systems. By the way, Apple, or Steve Jobs recently said in an email that Apple had no plan to do this kind of testing and signing on OS X. It's just for the portable apps. He says: That's not quite right. Symbian has for years enforced a process where applications have to be signed before a user can install them. When installing the application it tells the user what parts of the system will be used, such as the networks or the contacts list. Symbian is a open source OS and currently the most used in the world, about 45 percent of all smart phones sold.

While I do agree that the iPhone OS is a secure platform that would be very difficult to get arbitrary malicious code to run on, it would not be impossible at all to get trojan-like applications to run which, while appearing to be a useful application, were actually doing something you did not expect, like harvesting your contact list, for example.

My point is merely that scanning and signing a binary file does not guarantee security, nor must such benefit be exclusive to commercial operations such as Apple's Apps stores. Keep up the good work. Well, he does leave out one thing, which is that Apple approves all applications. It's not merely signed, but Apple has actually, presumably has some sort of testing process.

Steve: Well, yes. And so I think...

Leo: Symbian does not.

Steve: Right. And I guess Android does not, either.

Leo: Android does not, that's right.

Steve: So I wanted to say, and it may have been one of the other questions was going to prompt me, but I just sort of wanted to say that I'm not meaning to sound overly unambiguously bullish about the iPad. I'm excited about it because it's the first thing that I've ever had that gave me instant-on browsing and mobile PDF reading that was really practical. I don't, I mean, I would love HP's forthcoming Windows tablet to be really wonderful. But the battery life is half of what the iPad is, and you've still got to boot or hibernate and restore and all that nonsense. And the iPad is instant on and even receiving email when it's asleep.

So it may very well be - I'm quite fickle, as you are, Leo - it might be that there'll be an Android pad coming along that will yank me off of the Apple solution in a heartbeat. So at the moment Apple's the only game in town with this technology. And I have to say that there isn't, and I think this is where the critics have come down on the iPad, there isn't anything really breathtakingly spectacular about it. I mean, it's not like there's amazing new technology which is going to prevent anybody from coming up with a clone. And we know that the clones are on their way. If they're better in some way, but not worse in any way, that's what I'll switch to.

Leo: Yeah. Exactly. I think people often assume when we express, at least of me, not of you, but of me that, when I express a preference for an Apple product, that suddenly I am an Apple fanboy. But I should point out I use a Google phone, an Android phone. And I use what I like.

Steve: Yeah.

Leo: If I don't like something, I let you know.

Steve: So to Alex's point, I think that the more oversight that is brought to bear, the more security you will have. So, and it is the case that we've seen the difficulty of adding security after the fact.

Leo: Right.

Steve: So the fact that these new platforms are creating sandboxes and are so security conscious and are requiring things to be signed, that's just all good. And I'm glad that we're seeing an increase over time of security. Lord knows we need it.

Leo: Well, speaking of all good, thank you all for your questions. Future feedback shows, we do them every other episode. You can ask a question by going to GRC.com/feedback. GRC.com/feedback. Of course GRC's a great place to go for SpinRite, the world's best hard drive and recovery and maintenance utility. It's also the place to go for Steve's fantastic solutions, most of them free, like Wizmo, ShieldsUP!, Shoot The Messenger, DCOMbobulator and all the rest, the Perfect Paper Passwords. GRC, Gibson Research Corporation, GRC.com.

Steve has 16KB versions of this show there, along with all the other shows; as he said, transcripts, fully searchable; and show notes, too. GRC.com/sn. Steve, we will see you next week for more security. Do you know what you're going to talk about next week?

Steve: We're going to get back to the thread that we were on, talking about the - this is the "multiverse" was what I was calling the episode, multiple threads, multiple processes, multiple stuff. So a little more of our fundamentals of computing series.

Leo: Fantastic. I can't wait. GRC.com, ShieldsUP!, Leo Laporte, Steve Gibson...

Steve: SpinRite...

Leo: SpinRite. Have a great week, Steve.

Steve: Thanks, Leo.