SECURITY NOW!

Transcript of Episode #245

The Security of Open vs. Closed

Description: After catching up on many interesting recent security events, Steve and Leo seriously examine the proven comparative security of open versus closed source and development software, and open versus closed execution platforms. What's really more secure?

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-245.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-245-lq.mp3

Leo Laporte: This is Security Now! with Steve Gibson, Episode 245 for April 22, 2010: Open vs. Closed Security.

It's time for Security Now!, the show that covers everything you need to know about keeping yourself safe online. And the king of security is here, Mr. Steve Gibson, the man who discovered the first spyware, coined the term "spyware," wrote the first antispyware program. He's also the author of many useful security tools and the great SpinRite, the world's best hard drive and maintenance utility. And he's here from GRC.com to talk about security. Hey, Steve.

Steve Gibson: And not for the first time, for the 245th time.

Leo: I hate hearing those big numbers. They make me tired. I am tiring. So we're going to actually cover today something very interesting. A little more philosophical show than usual.

Steve: Yes. And I'm - I think that's precisely it. Last week when I was talking about the iPad it generated a flurry of responses from our listeners and some confusion over in GRC's own Security Now! newsgroup with people who were sort of assuming that I meant something that I didn't mean. And it sort of - and I ended up putting together a careful posting, sort of in reply, to make it clear what it was that I meant, but also to be a little bit controversial, and I think defensibly so. And we've never really addressed front-on the issue of open vs. closed from a security standpoint. We've gone around and around about what's more secure, open source or closed source? And so I want to talk about that, and also about open platform vs. closed platform, which is where the iPad and iPhone and other devices come in. So this week is the security of open vs. closed. And a bunch of news. It has been a hopping week. Got all kinds of really interesting stuff.

Leo: It's such a great topic. Yeah, good. Yeah, there has been. I think, well, I can't wait to hear what your thoughts are on it. And I will participate in this because I'm very interested in this, yeah.

Steve: I know you will. This'll be much more of a discussion between us. What I'm going to do is I'm going to - I've rewritten that original posting a bit for the podcast. So I'll lead in with reading what I wrote because I can't even paraphrase it as well as what I deliberately put down. And then we'll open it up to talk about it.

Leo: Great. Shall we start with the security news? I bet there's a little bit to talk about.

Steve: It was a hopping week, Leo. CBS did a report which was really rather scathing. You can find it on YouTube if you're curious. What they revealed in an investigative journalism piece was that nearly every digital copier sold since 2002, so that's the past eight years, contains a hard disk drive which, for reasons that surpass understanding, maintains an image of every document that the machine has scanned or faxed for you. And these machines are often leased by companies for some period of time. When the lease is up, they want the newer model.

Leo: Right.

Steve: So some guy comes in and wheels in the new one and wheels out the old one. The old one goes into a warehouse. And this CBS story showed a warehouse containing thousands of sort of wrapped-in-plastic, sort of shrink-wrapped copiers. At the time the story was being done, two large containers, as in shipping containers that go on huge transatlantic ships...

Leo: Oh, the freighters, yeah, yeah, yeah.

Steve: Yeah. Two of those huge containers were being filled up with copiers, headed for Argentina and Singapore. Now, the investigators purchased for, I think it was maybe three or $400 each, four of these used copiers and took them to a forensics guy, who used freely available software, opened them up, took out the hard drives. And they found literally tens of thousands of documents.

One of the copiers had been in a police station for the duration of its life, where it had scanned police records, driver's licenses, tax receipts, tax records, I mean, the list of documents that they recovered was mindboggling. One of the other of the four copiers had been at a health insurance company for its duration, where they recovered all the documents that the scanner had seen, which for some reason were stored dutifully on this hard drive. And they found reams of personally identifiable information, medical records, health records, social security numbers, I mean, everything you can imagine.

And so here was four randomly sampled copiers out of a huge warehouse. One of the employees during this story said, "Oh, yeah, we're getting copiers in all the time." And they said, "Well, where are those two containers going?" "Oh, one's off to Argentina, and one's off to Singapore. They buy our used copiers."

Leo: Hard drive and all.

Steve: Hard drive included. And one of the - I guess it might have been - it was someone, I think it was from Canon, was interviewed, one of the executives, who explained that, oh, well, yeah, we tell people, you know, that their documents are being stored in this thing. And for an extra $500 there's an encryption option. But apparently not everyone gets that. And it's not even clear what it would do or how it would work. And frankly, I'm mystified by what function it is that requires a digital copier to store documents.

Leo: Well, you could see there'd be short-term storage, maybe because it speeds up the scan, scans it in.

Steve: Well, exactly. And then you want, like, okay, we need 20 copies of this. So you feed it in once, it stores it on the hard drive, and then it dumps it out to its little, essentially a laser printer that's built into this thing. But have them expire after a day or after an hour. I mean, what's the point of storing every document you've ever scanned? And are they accessible through the control panel? I've not seen where you're able to somehow browse through past documents.

Leo: It's bizarre.

Steve: That would be an obvious security and privacy concern. So it's like this thing is hiding these documents for some reason. And anyway, so...

Leo: What about, like, Kinko's? If you use a public...

Steve: Yeah.

Leo: ...copy place, presumably that's all being stored, as well.

Steve: Yeah, I mean, I can't say one way or the other which technology does or doesn't. But, I mean, this seems to be a huge, unrecognized privacy problem where these copiers really do need to have their hard drives scrubbed before they leave, and of course that would kill the copier. So the idea is, when your lease is up, you don't want to...

Leo: Scrub.

Steve: Well, yeah, you can't kill the function of the copier. Presumably, if you did run DBAN or something against the drive...

Leo: Oh, I see, yeah.

Steve: ...it would render the copier nonfunctional. It probably boots off that hard drive, as well. So...

Leo: What's a digital copier? What is - how would you, I mean, how would you know if you had a digital copier? What is that?

Steve: It's probably any recent copier which is running a scanner and is able to do fancy things, like resize or duplex or uncollate, all those different things. You're going to have to store the scanned images somewhere. Clearly they're not storing it in RAM any longer. They're putting it on a hard drive. So, I mean, I would imagine that - and in the story I recognized these as WD hard drives, which are well known for many years as being very inexpensive in large OEM quantities. And so I could see the label on the drive going, oh, yeah, it's a WD hard drive, Western Digital. And those are in copiers, apparently. That's just the way they operate now. [Indiscernible] red flag.

Leo: That's very interesting. Very interesting.

Steve: In other disturbing news, Slashdot picked up a story from a security researcher, Kurt, I guess it's Seifried, who writes for Linux Magazine. He was curious in light of all of this recent flurry of concern over the security of certificates. He was wondering how difficult it was to obtain certificates for webmail systems because he realized that determining the identity of the person asking for a certificate is the remaining real problem with the whole certificate authority process. That is, when I apply for a certificate for GRC.com, one of the things that is done is an email is sent back to my address. And sometimes it's root@grc.com, or hostmaster, or webmaster, or postmaster, or something that they consider to be authoritative. They want to verify that somebody with a name like that at the domain that I'm getting a certificate for is able to receive the mail. And then they also will do, for example, a telephone loop.

Well, one of the problems that has arisen is, over time, the whois records, the domain registrar records, are being either spoofed or privatized, kept out of public eyes because people don't want people scanning the whois records in order to spam them. So protecting and making private your address is one of the things that many registrars offer. And some of them will simply just blank them entirely. They're just not available. So the problem is, how does a registrar verify the identify of someone requesting a certificate?

Well, the bad news is, what has come to light is that there are well trusted certificate authorities, like RapidSSL in this instance, who, if you are able to receive email from admin@, administrator@, hostmaster@, info@, is@, it@, mis@, postmaster@, root@, ssladmin@, ssladministrator@, sslwebmaster@, sysadmin@, webmaster@, whatever domain, that's all they require as proof of your identity. So the problem is, webmail hosting is @somedomain, wherever the webmail is hosted. For example, let's say Yahoo!. So unless Yahoo! has already locked down all of those names, if you can get an account as, for example, hostmaster@yahoo.com, or ssladmin@hotmail.com, then that's all you need to do in order for, with RapidSSL, a fully trusted certificate authority that all of our browsers trust. They will, and this guy did - I'm trying to remember the number now. It was either 11 different webmail companies, or he tried 11 and the majority, he was able to obtain from them to purchase a valid SSL web certificate, which we now understand allows anyone to perform a fully authenticated man-in-the-middle attack and/or to impersonate that web server.

So that's how weak the authentication security has been. And the problem of course is, for example, in the case of GRC, I control the GRC mail system. So no one is able to get an account on my domain. But the nature of webmail, and this is what makes it particularly vulnerable to this, is that you get an accountname@yahoo.com, accountname@hotmail.com or whatever, and these particular account names, in the case of RapidSSL, automatically qualify you to receive a certificate at that domain. Is that amazing?

Leo: So is this their fault? I mean, is it this particular domain company? Or...

Steve: Well, the problem is...

Leo: Anybody could do it.

Steve: Yes. He did this with, I believe it was 11 different webmail companies.

Leo: Oh, okay.

Steve: So this is a problem they all have. And, I mean, when you step back a little bit from this, it's like, okay, now, what should they do? Well, the problem is, there really is no - there's no one responsible here. They're wanting to sell certificates.

Leo: Right.

Steve: They're saying, well, how should we prove someone's identity? How does anyone know someone's actual identify on the Internet? With domain name records now being blocked, or being spoofed, or being deliberately obscured, we can't really rely on those. So we just sort of, if, you know, you give us your authentic email at that domain, then if we send you email there, and you're able to prove you got it, then we say oh, he must be the hostmaster, or the administrator, or the ssladmin or whatever. And they're saying, what else could we do? And so what this really does is it points out just, unfortunately, how bad the anchors are to the whole PKI, Public Key Infrastructure, chain.

Leo: Is it just RapidSSL, though? I mean, are other CAs...

Steve: No, there were 11 different...

Leo: Okay. So getting rid of RapidSSL from Firefox, for instance...

Steve: Well, I mean, he is proposing that, with there being a certificate authority as bad as RapidSSL - now, of course the problem is many legitimate companies have probably purchased their certs from RapidSSL.

Leo: Right, so you can't just dump them.

Steve: Exactly. Because then suddenly you're not going to be able to connect to their - to legitimate websites.

Leo: But other CAs aren't doing this? Just RapidSSL?

Steve: It's not clear.

Leo: Okay. He tried it with RapidSSL.

Steve: He tried it with RapidSSL and recognized that this is a problem that any certificate authority would have. And so he was able - so his contention was, what you need to do, I mean, what webmail companies should do is absolutely acquire all of those account names so that a company like RapidSSL, who is wantonly distributing certificates, won't distribute any for your domain. That is, you control those, sort of those special account names. Now, the problem is, different certificate authorities have different account names that they consider qualifying for that domain. It just looks like RapidSSL has a large collection of them.

Leo: Wow. So I own a few domains, like TWiT. Should I then do something to protect myself against somebody getting administrator@twit.tv? Do I have to register, I mean, it's an infinite number; right? I mean...

Steve: Well, except that how can some random person get an email account at twit.tv? I mean, that's...

Leo: Oh, they can't. So that's why Hotmail works or whatever, because you can get leo@hotmail.com and then register administrator@hotmail.com.

Steve: Exactly. It's why you and I are safe is I control my email system; you control yours.

Leo: Okay. Now I understand. Got it.

Steve: Right. And so the problem, this particular problem that affects webmail is that these special account names can be acquired on webmail systems. And that's all that a company like RapidSSL requires in order to say, oh, you must be the domain owner. We'll give you a certificate. And one of the things they ought to do is they ought to check to see whether that domain already has a certificate. All they have to do is connect to its web browser, and it'll say, oh, yeah, the actual domain has a certificate from VeriSign, for example. So why is it you're getting a certificate now from us? Oh, and that certificate's got three years left of life on it or something.

So, I mean, there are more proactive things that they could do. But the problem is there's no accountability. Nothing holds a certificate authority accountable for the certs that it's issuing except, if it gets a black eye, like by being way permissive, and people start removing that CA from their browsers because they no longer trust that CA, then people are going to be disinclined, valid webmasters are going to be disinclined from purchasing certificates from them because people have removed the root CA from their browser, preventing them from getting to their server securely.

Leo: Have you been following the McAfee story that broke this morning?

Steve: No.

Leo: Oh, you're going to love this one. McAfee released a update to their Total Protection Antivirus, I think it's actually to the corporate version, that identifies svchost as malware and deletes svchost.exe.

Steve: No. How can that possibly be?

Leo: Then sets off a chain of uncontrolled restarts and loss of networking functionality. Twitter's going nuts about this right now. Engadget has a statement from McAfee that they've pulled the update from their corporate download servers and that consumers shouldn't be affected.

Steve: Oh, their corporate servers?

Leo: It's corporate.

Steve: Ohhhhh.

Leo: It's DAT update 5958. Here's the release from McAfee. McAfee is aware that a number of customers have incurred a false positive error due to incorrect malware alerts on Wednesday, April 21st. That's as we record this. The problem occurs with the 5958 virus definition file that was released at 2:00 p.m. GMT, 6:00 a.m. Pacific time. Our initial investigation indicates this error can result in moderate-to-significant performance issues on systems running Windows XP SP3. Apparently it affects SP2, as well. The faulty update has been removed from McAfee download servers for corporate users, preventing any further impact on these customers. We're not aware of significant impact on consumer customers and believe we have effectively limited such occurrence. Nobody knows. The anecdotal numbers, according to Engadget, are 30,000 to 60,000 machines rebooting, as we speak. in a loop, an endless loop.

Steve: Well, and Leo, without the svchost.dll, that's a key DLL used...

Leo: You can't run.

Steve: No. I mean, Windows won't run without that. You might as well just, you know, reformat the hard drive.

Leo: The fix, if you're listening, is to boot to safe mode, rename McShield.exe, the McAfee Shield, McShield.exe, reboot, run Virus Console, pick Tools, and roll back the DAT, go back to an earlier antivirus definitions update. And then you can restart McShield and reboot. That's...

Steve: Interesting.

Leo: I don't know if that works or not. That's from Lifehacker. They say it's an unverified tip.

Steve: Wow.

Leo: Whoopsie.

Steve: Yeah. When good software goes bad.

Leo: Well, I'm not sure I'd call it good software.

Steve: Yeah, ooh.

Leo: Might be an overstatement.

Steve: Ouch. We talked several times now about the problem with PDFs being able to execute code. So I wanted to alert everyone that the well-known and three-year-old Zeus botnet malware is now apparently active in about, so it's been reported, as much as 88 percent of the Fortune 100 companies because it is infecting PDFs. When users open an infected PDF, they're prompted to save a file called Royal_Mail_Delivery_Notice.pdf, which is actually a malicious Windows executable. The trojan installs a sophisticated, difficult-to-detect and difficult-to-remove keystroke logger which steals login credentials to banking, social networking, and email accounts.

The command-and-control servers have been located, but they're well distributed, so they've found a few of them. And they have been found to contain tens of thousands of pieces of personally identifiable data, credit card information, social security numbers, login data to banking accounts and email account info. And it's believed that several million machines worldwide are currently infected with this. So I did want to remind our users to do what Adobe is now saying to do, which is turn off this ability for PDF files to run executables. It is now being actively exploited in the wild, I mean, as it was inevitably going to be. And it's catching a whole bunch of people.

Leo: Yikes.

Steve: Also, as we were recording last week's podcast, and I was sort of lamenting, remember you and I were discussing how prevalent the Java Runtime was in people's machines?

Leo: Yeah.

Steve: As we were recording that, or I guess it was the day after we were recording that, Oracle, that of course owns Sun, was quickly putting out an emergency Java patch to deal with this zero-day vulnerability which they had previously decided was not serious enough to warrant an out-of-cycle patch. They were going to wait until July.

Leo: Oh. July?

Steve: July was when this was going to get fixed. But then it became clear that it was already being exploited in drive-by attacks. For example, users who visited SongLyrics.com would find their computers compromised if they had the Java Runtime, which would be - the vulnerability in it was being invoked by scripting on the SongLyrics.com website in order to install malware into their machines. So Java is now at, as of last week, Java 6 Update 20. And Oracle decided this thing would not keep for a few more months. So...

Leo: Really. Oh, that's good.

Steve: That's good news. And then over in my - my favorite acronym is TNO, and in this case I realized that you could rearrange the letters and say TNO NOT. We have Wired.com reported...

Leo: That's the Yoda version. No One Trusts.

Steve: Exactly, the first publicized cloud computing warrant. Wired's story said "Spam Suspect Uses Google Docs, FBI Happy." That actually does sound a little bit like Yoda.

And I'll just read from this because they - rather than paraphrasing. It says, the beginning of the Wired.com story: "FBI agents targeting alleged criminal spammers last year obtained a trove of incriminating documents from a suspect's Google Docs account, in what appears to be the first publicly acknowledged search warrant benefiting from a suspect's reliance on cloud computing." Thus Trust No One.

"The warrant, issued August 21 [of 2009] in the Western District of New York, targeted Levi Beers and Chris de Diego, the alleged operators of a firm called Pulse Marketing, which was suspected of launching a deceptive email campaign touting a diet supplement called Acai Pure. The warrant demanded the email and 'all Google Apps content' belonging to the men, according to a summary in court records.

"Google provided the files 10 days later. From Beers' account, the FBI got a spreadsheet titled 'Pulse_weekly_Report Q-3 2008' that showed the firm spammed 3,082,097 e-mail addresses in a single five-hour spree."

Leo: Oh, please.

Steve: "Another spreadsheet, 'Yahoo_Hotmail_Gmail - IDs,' listed 8,000 Yahoo webmail accounts the men allegedly created to push out their spam. The Yahoo accounts were established using false information, allegedly in violation of the CAN SPAM Act." So here's - we've talked often about what it means to have your documents stored in the cloud in, unfortunately, a nonencrypted fashion. I've got a bunch of stuff stored in the cloud, but I'm doing it through Jungle Disk, where Jungle Disk itself performs an AES 128 or 256, I can't remember now, but very strong AES encryption of everything leaving my machine so that, even though Amazon is storing the data, what they're storing is gibberish. They're storing pseudorandom content which, under no compulsion possible can they reveal anything other than pseudorandom data. So, I mean, this is something our listeners just need to be aware of. I was listening to you, Leo, on one of your other podcasts, talking about Google Docs and hearing some of your listeners, I mean, your...

Leo: Yeah, we use it like crazy. I mean, this is our new way of - in fact, you know, we tried to send you a Google Doc. Yeah.

Steve: And for things like show notes and so forth.

Leo: Show notes, we keep track. But all of our, to be honest, all of our company stuff is on Google Docs.

Steve: And so, again, as long as that stays secure, as long as you're not concerned about the possibility that a legal warrant could be presented forcing the disclosure of all that, I mean, you need to understand, I mean, anyone, my point is anyone doing this has to understand that their data is not entirely under their control any longer. And it's not that they only need to trust Google. They need to trust the entire chain of the federal government that has the right to issue warrants to compel somebody storing this information to release it.

Leo: Right.

Steve: So it's worth being aware of that. Now, unfortunately, we have a related story, which is that Network Solutions' Unix-based webhosting servers have been breached yet again. We haven't talked about this before. But it's a problem that has been continuing. Network Solutions had a problem just a few weeks ago, and then a few months ago. In this case, hundreds of their hosted websites were hacked. And they're being rather closed-lipped about it at this point. I have a quote from them saying "We have received reports that Network Solutions customers are seeing malicious code added to their websites, and we are really sorry for this experience."

Leo: Yeah, I bet they are.

Steve: The code, which has been injected into hundred of Network Solutions' customers' web-hosted websites, redirects any visitors to their sites to more deeply hacked servers that silently attempt to install malicious software using a variety of known vulnerabilities such as those that exist in Adobe's PDF Reader and insecure ActiveX components.

So here's sort of a different take on this. And that is that we can hope that Google's systems are secure. On the other hand, we know that just a few months ago there was the whole much-publicized infestation of malicious activity in Google's network, allegedly from people in China that had been inside of Google's network and the networks of many other major corporations for some number of months. So TNO.

Leo: TNO. Or NOT for the Yodas among us.

Steve: In errata, we have some clarification of the problem that I also heard you discussing. I guess it must have been MacBreak Weekly you were doing, I think it was yesterday, talking about the problems with the iPad and networks, that...

Leo: Yeah, DHCP lease issues.

Steve: Yes, exactly. There was news early this week that major East Coast college and university campuses were banning the iPad, or the iPad was causing all sorts of havoc...

Leo: Princeton did, yeah. And George Washington wants to, and another one.

Steve: Yeah. And it turns out that we now understand - there's been some very good posts by Cornell, I think it was, that I guess due to the nature of the oversight that they provide for their own DHCP servers, they were able to relatively quickly figure out what was going on, which is that the iPad makes a mistake in not ever renewing its DHCP lease which has expired, if the expiration occurs while the iPad is sleeping.

And what I didn't realize until I read this detailed report is, when you press the Power button, what looks like the Power button on the iPad, you're actually just putting it to sleep. And I should have understood that because I've noticed, for example, that it will, if it's plugged into my Mac, being recharged, it'll "bong" when it receives new incoming mail. And also I made a comment, also last week, about how - or maybe it was the week before - how it's automatically connected to networks that I am at, such that what I really wanted was instant-on web surfing, and it gives me that because, when I turn it on, I notice that it's already connected. So that's not really off. It's sleeping.

And so this is clearly something that Apple will fix, hopefully very quickly. I'm sure they're on top of it now. But the idea is that, as our listeners probably know, when you obtain information from DHCP, Dynamic Host Configuration Protocol, over a network, it's providing you with things like the DNS servers and the IP of the machine and any of a number of different pieces of information can be obtained from the DHCP server. And the idea is that there's some expiration on that. There's a "lease," as it's called. So you're merely leasing this information for some length of time. And what should be done is that, at some point prior to the expiration of that, the client of the server that has received this information will renew or refresh the lease, normally keeping the same information, but just sort of saying, hey, I'm still here, still using this. I want to keep this IP allocated to me.

Leo: So this only happens when the iPad auto-locks. As long as it doesn't auto-lock, no problem.

Steve: Exactly. When it goes into auto-lock mode, for whatever reason, it puts it in a state where, if during that time the DHCP lease expires, then the iPad is unaware of that happening. If you then wake it up again, with the lease having expired, it does not renew the lease. So what was happening in the case of Cornell University is that iPads were auto-locking during the time that the lease was expiring. Then the users would turn them back on again.

Well, if that IP address which had been assigned to that iPad had subsequently been assigned to somebody else, then you'd have two machines with the same IP address on the same network, and we know that's never good. So you get an IP collision in that case. ARP isn't happy. You've got different MAC addresses assigned to the same IP. And indeed it will cause a problem.

So what Cornell was doing was, initially, if this happened twice, they would blacklist that device by its MAC address, which essentially kicked that device that was misbehaving off of the Cornell network. When they later figured out exactly what was going on, they arranged to inform iPad users of several different workarounds which could be performed, one being that you would disable the auto-lock so that you'd have to manually put the iPad to sleep, which prevents this. And before doing so, they suggested you also manually turn off the WiFi. And again, so students are having to jump through these hoops until Apple comes up with a fix, which I imagine will be forthcoming pretty quickly.

Leo: Any minute now, I would imagine.

Steve: Yeah.

Leo: Yeah. So it's a legitimate complaint on Princeton's part.

Steve: Oh, absolutely. This is very bad behavior from a client. It's doubtless some little tiny mistake which occurred due to the implementation, some implementation detail, this particular state that the iPad goes into that causes this problem.

And I do also need to explain to my listeners that, when I talked about how gleeful I was that the iPad was connecting to open hotspots, I didn't mean that it would ever do that to hotspots that I had never been to before. It behaves just like Macintoshes and Windows machines do. That is, you tell it that you want to trust this hotspot, and so that trust is persistent, such that if you return the next day to a hotspot that you had told it to previously trust, and you've given it permission to remember that, then it will connect. It's not as if it's running around promiscuously connecting to random hotspots that you've never connected to before.

And a lot of people in the newsgroup and also who were sending email said, wait a minute, you think that's secure? That sounds incredibly insecure. And it's like, no, no, no. That's not what I meant at all. So it behaves in the same way that our existing laptops do, in just that they're remembering the hotspots that they have visited before.

Leo: So it really isn't an oversight on Apple's - well, it is an oversight on Apple's part, but it's kind of an understandable oversight. It's something they may not have thought about. Because they're acting like a laptop.

Steve: Yeah, exactly. No, in this case it's behavior that you want. You want to be able to say this is a place I go to, this is a coffee shop I go to, so trust its network because I'm trusting it now. I would like you to remember it.

Leo: Continue to do so, yeah, yeah.

Steve: So, yeah, so that I trust it when I'm back there tomorrow. But if you go to somewhere else, it'll say no Internet connection. Then you go over, go to the Control Panel. You open up WiFi. And it'll show you a list of available networks which may or may not be secured. And so you make your decision, and then you tell it whether you want to remember that in the future. Just like Windows or Mac works.

Leo: Right.

Steve: I did have a nice story from Robin Weber, who is a listener of ours. He wrote to say that SpinRite had saved five generations of our family.

Leo: Wow.

Steve: He said, "Thank you, thank you, thank you, Steve. Two months ago my wife's PC stopped booting. It would actually go into a reboot loop after the XP loading screen, with an ugly sound coming from the hard drive before each reset. I tried several recovery tools on the damaged drive mounted in a good PC and never got back anything more than the sounds of mechanical gagging.

"Two weeks ago I stumbled upon the Security Now! podcasts and started listening to them. Then halfway through my first podcast I heard the word 'SpinRite' mentioned. I thought that name sounds very familiar. Could it be the same software that I used to use over 10 years ago for drive recovery? Anyway, I knew I didn't have a hope of recovering the drive unless I could get Windows out of the way. So I put down my $90 U.S." - actually it's $89 - "for SpinRite and downloaded it. It was a 170K file. I thought it had just downloaded a loader program..."

Leo: It was an error, yes.

Steve: "...which would then get the rest of the program from the Internet. But no, $89 U.S. for a single 170K file. But getting out of my stupid Windows mentality, where you need a gig of free space just to compile a program, I thought, why the hell should it be any bigger? So I ran it, and it wanted to create a boot floppy or an image which you could burn into a bootable CD. So what the hell. I've just spent all this money, may as well give it a CD. 40 minutes later, after running SpinRite on Level 2, I had the entire disk back."

Leo: Oh, wow, that's great.

Steve: "Didn't lose a single magnetic bit. Well, there were a few early sectors reported as partially recovered, so I probably lost some innocuous files. But my wife is over the moon, as she's got all of her gen- genol-..."

Leo: Genealogy.

Steve: Thank you, "...all of her genealogy database plus all of her photos, emails, et cetera. And the kids are happy because they have their saved games back. We've now backed up the disk, and I'll replace it shortly. I just spent $89 and got two years' worth of rework done, well worth the cost. Thanks again, Steve."

Leo: Isn't that great.

Steve: And thank you, Robin, for your great report.

Leo: That's so great. Well, Steve, we're going to get to the meat of the matter, open vs. closed systems and which is more secure. A great conversation. I can't wait to this. All right, Steve. On we go.

Steve: So I will lead in by reading an edited version...

Leo: Good.

Steve: ...of the post that I wrote for the newsgroup because it says it better than I could paraphrase it. And then we'll talk about it. So, open vs. closed. First off, there are very important differences between open vs. closed code and development and open vs. closed execution environment platforms.

First, looking at the code and development methodology side, I'm not at all clear that open source code is inherently more or less - either way - secure than closed source code. I don't see anything other than personal policy bias, or commercial interest, to recommend one over the other from a security standpoint, based upon all of the history we have with the demonstrated security arising from either approach.

We have a great pair of samples in Microsoft's Internet Explorer, which is of course closed, and Mozilla's Firefox, which is famously open source - the first one as closed as it could be, and the second wide open to the world. Yet over the past few years, as Firefox has become more prevalent, we don't see significantly fewer problems on that platform. We may see more active exploits of IE due the fact that it still retains a huge majority of web browser market share, and the fact that it's also always installed in every machine, and also the fact that it's the more cautious and careful security-aware users who have taken the trouble to run Firefox and then add on the additional controls for cookies and scripting management. But, overall, looking at the logs of problems being fixed by the now continual flow of Firefox patches, recently more than weekly, we're not seeing anything that says open source means more security.

In this weekly podcast I am forced to skip over reporting on the mass of security problems continually being discovered in open source software because otherwise we would never have time to talk about anything else, and because they generally are spread out among a great many different pieces of software rather than focused, and in general they have a low level of saturation per listener.

>From all of the wealth of evidence we have seen, I think that in either case of open or closed source software and software development, it is the resulting delivered product of either approach, which is pounded and pounded upon and never needlessly altered, except for the purpose of carefully fixing small implementation errors, that over time earns the right to be called "secure."

Our listeners may recall my annoyance at Microsoft's Steve Ballmer declaring, at launch, that Windows XP was the most secure operating system they had ever created. And my comment back then was, "That's not something that anyone can simply pronounce. Security can only be proven over time through being tested."

The security of any given implementation of a system can only be earned. Security cannot be "declared," it must be proven. And establishing that "proof" requires time spent in the line of fire. And this is easily demonstrated by the simple fact that it's entirely possible to create vulnerable and exploitable code under either type of source code model, either open or closed. The whole nature of "debugging" is such that programmers will miss both their own and others' coding errors since they get sucked up into making the same assumptions.

And it really doesn't matter how many eyeballs examine the code. Closed source economics at Microsoft can certainly afford to employ just as many eyeballs looking at code as does the volunteer open source model. And anyone who has ever actually been involved in open source projects knows that, in reality, only a very few, and oftentimes just one, developer is actually doing most of the heavy lifting on major parts of a project. So I don't buy for a moment that there's any intrinsic benefit either way in open vs. closed source. Either can be buggy as hell, or stable and solid as a rock.

And it's worth noting that, in the case of closed-source code, commercial interests work both for and against its security. On the one hand, commercial interests need a reputation for security to help bolster sales; and, on the other hand, commercial interests are motivated to keep adding features and revising perfectly good and previously proven code in order to keep their users upgrading and revenue flowing. No, mistakes can be made in either development environment, so I see no benefit there, either way.

But what really has a profound effect on security is policy. And this is where massive differences in security results can be obtained. Years ago, how many times did our listeners hear me grumble, not about the fact that a defect was found in Microsoft's Outlook that allowed email to take over the user's machine; but that Microsoft continued, year after year, their policy of enabling any scripting at all in email. Mistakes are going to happen, but policy is deliberate. And policy is directly reflected in design.

Which brings us to the question of open vs. closed platforms, which is an entirely different animal from open vs. closed software development. In comparison to an inherently closed platform, which deliberately imposes restrictions upon what software is allowed to run on it, and upon what freedom permitted applications have, any inherently open platform that permissively allows anything to be developed for it and run on it is going to, by design and definition, be significantly less secure. The more "freedom" applications have within a platform, the less secure it will be, since freedom is so easily abused. And the more inter-application interaction the platform allows, the less secure that platform will be, since so many games can be played between applications.

And any platform whose design fundamentally adopts an untrusted application model, assuming that applications may misbehave, and by design policy strictly limits their freedom - as, for example, in the case of the iPhone OS with its application sandbox - versus any platform whose design is fundamentally one that encourages a community of assumed mutual trusting applications, as Windows, Unix, Mac OS X, and other "traditional" open platforms do, will be inherently more secure by design.

And, finally, the more "locked down" the application screening process is, the more difficult it is for applications to be approved, as individual applications are examined by the platform's administrator scanning for the use of undocumented APIs and any other possible misbehavior, the more inherently secure that locked-down platform will be.

Thus, taking Apple's just-released iPad as an example, while we cannot possibly say today that the iPad - a three-week-old product when we're recording this podcast - is secure because by definition that can ONLY be proven over time, we can definitely state that the iPad's fundamental design, by virtue of the deliberate and often infuriating and disappointing limitations that were designed into it from the start, make it as a platform not only fundamentally more secure, but also fundamentally more securable. So there.

Leo: Well, you make a very interesting distinction, which I think is very important, about security policy versus - it's not really merely open. It's the policies. Because...

Steve: Right.

Leo: And the reason - at first I thought, oh, now, this will be interesting to see whether Steve makes a distinction between a closed - between open source software and an open system versus a closed system. Because, I mean, it really isn't merely, I mean, some of this software on the iPad is in fact open source software. But that's not really what makes it secure, it's that it's a closed system. It's the policies; right?

Steve: Right. In fact, I spent some time this week delving into the developer documentation for the iPhone OS because my own curiosity was raised by this question. And what is very clear is that Apple designed a deliberately limited system. They explain to the developers that users need to select an app, that the OS will run that app, that it will have the entire screen while it's running, and that when the user presses the Home button, the application will be told to terminate. It will have five seconds to do so. And this is where we're talking about existing iPhone 3.1.3, the existing level, not the forthcoming 4.0 that begins to allow a little more leniency in termination. But the idea is that, when you press the Home button, the application is told to shut down.

Leo: You're done, yeah.

Steve: And it's got five seconds to do so. And if it doesn't, it will be preemptively terminated by the underlying kernel. And in fact the developer docs explain that users would like to have a feeling of continuity from one instance of running to the next. So it's up to the individual application to use that shutdown time to quickly save its own state so that, when it's run again, it has the option, and hopefully will take advantage of it, to look like it kept running, even though it didn't.

And so this means a number of things from a design standpoint. It means that the application has the whole phone to itself while it's running, so there is not any complex multitasking necessarily going on, although the system itself is multithreaded and can be doing many things at once. But there's one task that has essentially focus, in that it has the whole screen, and that it has all of the resources of the machine at that time.

What's interesting also, Leo, is that when an iPhone application is installed, it's given a unique token which is the name of the application's file space, so that there's even this - the application essentially can have its own storage area. But there's absolutely no visibility into the rest of the system. There's no file system that the application can rummage around in. It has no visibility at all into the file system that other applications are sharing. So from the start there was this notion of separation of processes running on the phone. Now, this makes sense in something you have in your pocket. None of us would put up with this on the platform that we've got on our desktop. Windows and Mac, we're used to huge screens, lots of things going at once, jumping around between things.

But so it's worth also noting that I talked about criticiz- or I did criticize Microsoft for having scripting, leaving scripting there in Outlook for so long, I mean, year after year, where this posed horrible problems for security. The reason they did is that they just - nothing can make Microsoft take out a feature because, even though I don't know anybody who ever used scripted email, there were probably some companies in the Fortune 500 that were using it; or the fact that the feature was there, Microsoft was terminally afraid of breaking something by removing a feature.

So the problem is, and we've seen this with the evolution of Windows, how difficult it has been over time for Microsoft to try to get back security policy that they had originally never considered. Things like UAC that appeared in Vista. Things like DEP, the Data Execution Prevention. I mean, that stuff sort of creeps in slowly. It's there, but it's turned off. And then it's kind of turned on, then it's turned on a little more. And then finally it's on by default. I mean, it just - Microsoft is trying to creep forward.

Well, so it's a huge advantage for someone like Apple to come along much later in the game, having watched all of this bloody war in the past, and said, okay, we understand what we need to do to create a fundamentally secure platform through policy. And developers aren't going to like it. And to some degree even the users are not going to like it. I mean, I was never able to get my iPod Touch on my WiFi here at home because I use one of my own impossible-to-type-in passwords. And so I couldn't - there was no cut-and-paste as we remember because that was a form of inter-application mischief that might have been possible. And, I mean, Apple's design decisions were so rigorous in the beginning that they said, if anything, we're going to err, on the iPhone, in favor of being overly restrictive because we can always loosen it later. But we can never tighten it later. And that's, of course, the lesson that Microsoft is still trying to learn and having to pay for.

Leo: Yeah. Start tough.

Steve: Yeah. And then, as it's proven, as you're able to - I have no doubt that there's some serious vetting going on about the data on the clipboard when you cut, copy, and paste among applications on the iPhone and the iPad because, again, Apple recognizes that what they have, they have established a strongly sandboxed, seriously secure policy.

And I've got to say, Leo, given my choice, I would rather go without features and not compromise security than have the iPad turn into a much more open platform that starts having real problems. Because this all came up with a Q&A question we answered last week, the guy saying would the iPad make a reasonable dedicated platform for secure banking. And I say there's the reasoning for my having said yes last week.

Leo: Yeah. I mean, it's not that you couldn't have an exploit on there. It's just that an exploit would be, because of the nature of the platform, kind of - its impact would be dampened. Is that it?

Steve: Well, and it's that, with Windows, it's almost impossible not to have exploits. I mean, it's completely open. It's complete - it's the Wild West. It's you go to a website, and something is installing code on your machine that runs. I mean, that can't happen on the iPhone because the iPhone is - you have to have signed applications that are cryptographically signed before the system will run those things. There's nothing like that on Windows. And you can't impose it now. It's too late.

Leo: It's too late, yeah.

Steve: Exactly.

Leo: As we talked about last week, Apple kind of - this might have been a stealth, intentional stealth act on Apple's part, that they thought, well, if we're going to start from scratch, maybe we start from scratch in a different way. And forcing signed applications, controlling tightly the application store, controlling how applications are developed, all of these things might annoy developers and end users. But it does have an interesting impact. And as I mentioned last week, if security becomes a bigger and bigger issue down the road, it might be interesting, it might be an interesting advantage to Apple that nobody had really thought about. I wouldn't use the iPad as my sole computer. I don't think - or iPhone. I don't think anybody is suggesting that. But...

Steve: Oh, absolutely not. I'm with you. I mean, it's with me when I'm out reading and surfing. And I'm wishing that it had things that I have on all my other real machines, like a really good newsgroup reader that will hopefully come along. But for what it is, and this is exactly our listener's question last week, I would argue it is the safest way to surf the 'Net today. I can't think of a safer way to surf the 'Net than on an iPad.

Leo: Yeah. All right. Well, very interesting points.

Steve: And it's worth also talking about the other sandboxed platforms. I mean, we do have Droid, for example.

Leo: Would the same thing, same case be there, you think?

Steve: Well, the problem there is that, I mean - and again, I'm not saying there's anything wrong with this. But Apple has obviously a strong economic reason for also controlling the apps. They want to be in the revenue stream of apps which are non-free which are being installed on their iPhone OS products, the iPhone and the iPad. Whereas the model that Android has is strong from a security standpoint, because again they had the advantage of coming along later in the game, but anyone who wants to can create apps for it because their economic model is entirely different from Apple's.

So while I would argue that, yes, it's better than the old, inherently trusting platforms, like Windows and Unix and Mac OS, it is because anyone can create an app that won't go through some sort of scrutinizing process. I mean, I would argue that you're getting something, there's some value from a security standpoint that you're getting sort of unwittingly from the fact that Apple is so concerned about the apps that run on their platform. They do scan them for the use of undocumented APIs. They scan them to the limit of their ability.

And I don't say that this is something you can feasibly do in an automated fashion. Obviously they're not doing a source code review or anything. They're just looking for, where they can, for behavior that looks suspicious. But it's only after it goes through that then that they will sign it to give it the cryptographic credentials that it needs in order to run on people's phones. There's some value there that you're paying something for, that Apple is getting some money for, that is not part of the economic model, the ecosystem in the competing open phone platforms. And so, yes, it's not free, but I think there's some security benefit, as well. And I'll get you that gets proven over time.

Leo: Yeah, we shall see. I mean, there's nothing to say that somebody couldn't develop an exploit for it. I mean, there are exploits for the iPhone. Not very effective exploits, but they're...

Steve: Well, and they all involve jailbroken iPhones; right?

Leo: Ah, that's interesting.

Steve: Yes. I mean, so when you do that, all bets are off. You've broken a chunk of the security of your system when you jailbreak your phone. So it's hardly surprising that, very quickly, that got taken advantage of. But as long as the security is up and intact from the kernel, and you haven't done something to deliberately weaken the security of your phone, it seems very strong.

Leo: There was a - at the PWN 2 OWN contest at CanSecWest, someone was able to break into a fully patched iPhone and hijack the SMS database, including messages that had been deleted.

Steve: Yeah. There are APIs which allow applications to have access to the contacts. And there are the problems also with - I know the hack you're talking about. It was a...

Leo: It used a Safari vulnerability, I think.

Steve: It was a return - it was using existing code, jumping into existing code in order to get the code to execute. And we've talked about this kind of approach before.

Leo: They say this exploit doesn't get out of the iPhone sandbox. You're still in the sandbox. The problem is, of course, that Apple's own applications have more access to the system than third-party applications. So if you can find a flaw in Apple's applications, in this case in iPhone Safari, you get farther than you would if you hacked Leo's application.

Steve: Right. And I guess my response is that we know that security is not absolute. It's not a matter of is - and I'm not saying it is impossible to hack the iPhone or iPad. Not at all. Again, that you can only get - that we can only prove over time. But it is clear that, by policy, there's a huge set of policies that are inherently restrictive and inherently good for security, and that that matters.

And so, I mean, if you look at - the security under Windows is a catastrophe. I mean, it's a disaster. I mean, it's horrific. And we see nothing like that on the iPhone after its three years of existence and it being very popular, and 50 million of them are out there. And it's a connected, communicating device that is fundamentally a fertile medium for this kind of attack. And it's quiet by comparison. And that's not a mistake.

Leo: Right. Great, great conversation, Steve. Fascinating stuff. Is that essay on the website now, or is it going up later?

Steve: I just wrote it.

Leo: So it'll be part of the show notes...

Steve: I'll email it to you so you can add it to your show notes.

Leo: Good. And you'll put it - will you put it on the website, or - well, it'll be part of the transcript. What am I saying? Of course it will.

Steve: It will be part of the transcript, and I will also make - I'll stick the file on GRC's show notes, so people can have that, as well.

Leo: You can get those notes by going to GRC.com. There's a whole Security Now! page with notes for all of the shows, including 16KB versions for people with less bandwidth. There's a PDF, there's a transcript, there's everything you'd want, all available at GRC.com.

When you're at GRC.com, take a look at all the great other stuff that's there. Steve has put together, of course, the world's best hard drive maintenance and recovery utility, SpinRite. There's no better way to do it. And you can find out all about that there. But also lots of free stuff on his site, including ShieldsUP!, Shoot The Messenger, Unplug n' Pray, the DCOMbobulator, Perfect Passwords, and my favorite little doohickey, Wizmo. One million downloads on Wizmo. That's pretty good.

Steve: Yay. They're getting there.

Leo: Yeah. Of course you've got almost three million on Unplug n' Pray. Wow. Wow. All at GRC.com, the Gibson Research Corporation. You can watch this show, we do it live every Wednesday afternoon at 2:00 p.m. Pacific, 11:00 a.m. - I'm sorry, 2:00 p.m. Eastern, 11:00 a.m. Pacific, that's 1800 UTC - at live.twit.tv. You can watch video on YouTube at YouTube.com/twit in the Security Now! channel. You can also download audio and video from any podcast aggregator, including iTunes, the Zune Marketplace, on Linux, everywhere that you can get a podcast. We have large and small versions available of the video, and a 64KB mono version of the audio. All for your delectation, absolutely free. Because we have support from great sponsors like Citrix, the folks who do GoToAssist; and Carbonite, who do Carbonite's great backup software. We thank them for making this possible. And I thank you, Steve.

Steve: And I do want to solicit feedback on this topic. I'm sure that a lot of listeners have opinions. GRC.com/feedback will get you to a web form. Say something about iPad or iPhone security in the subject line. And next week is our Q&A. And I have a feeling I'll be reading some opinions about this.

Leo: I bet, yeah.

Steve: I'd love to have them, either way.

Leo: We can talk about it. GRC.com/feedback to leave feedback for our talkback issue next week. Thank you, Steve.

Steve: See you, Leo.

Leo: See you next time on Security Now!.