SECURITY NOW!

Transcript of Episode #295

The Comodo SSL Breach

Description: After catching up with the past week's very busy security news, Steve and Leo closely examine the circumstances and repercussions surrounding the mid-March breach of the Comodo SSL certificate authority certificate signing system.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-295.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-295-lq.mp3

Leo Laporte: This is Security Now! with Steve Gibson, Episode 295, recorded April 6, 2011: SSL and Epsilon Breaches.

It's time for Security Now!. We're starting a little late today because of a scheduling change earlier in the morning. We had a great triangulation with Cory Doctorow, who is a fan of this show. Hello, Cory. He was talking about listening to this show. Steve Gibson is...

Steve Gibson: Oh, cool.

Leo: Yeah, isn't that cool? Steve Gibson is here. He's our security guru.

Steve: What a surprise. I'm here for Security Now!.

Leo: Wow. That would be the 295th time you've done that.

Steve: Who would have guessed. Yes, indeed.

Leo: Episode 295. We are going to talk about two big stories in the news today, the SSL breach...

Steve: At Comodo.

Leo: ...at Comodo, and the Epsilon hack, which is just breaking this week.

Steve: Well, and there's, I mean, there's even, like, you probably heard about this massive SQL injection which has affected millions of links that Google turns up. Just it's a huge automated SSL injection attack which is routing people to sort of mid-tier companies' websites to a malicious, your computer has been infected, download this to fix your problem kind of. So we've got lots of news.

Leo: I'll tell you how I always find out about stuff like that. Now, I had read about that last week. But of course I got a million phone calls on the radio show. Well, not a million, but three or four from people who had gotten bit by it. If I get three or four on the radio show, you know it's very widespread. So I'm glad we can cover that, as well.

Steve Gibson, the man at GRC.com, creator of SpinRite, security guru, author of a great many free security utilities at GRC.com. Let's get to the security news because there was another big story we didn't mention.

Steve: Oh, well, there's a bunch of stuff. We've got to go back a little bit and just sort of mop up the debris from the RSA big SecurID break-in that we talked about, I guess it was just last week. Some new information has come out from RSA. They're trying to paint it as a very sophisticated attack. Now, remember, this again, this is my most recent, my own personal blog posting, because what they initially announced and said was so annoyingly little that no one knew what to make of it. Yet they're, like, the major multifactor hardware token provider for the industry. As you mentioned, lots of government and Fortune 500 companies use RSA because, I mean, RSA are the inventors of crypto. State-of-the-art asymmetric crypto came from the founders of RSA.

So the fact that they were being so circumspect led us to believe, okay, well, sure, they're embarrassed, they may have some fiduciary obligations to their stockholders not to say too much, they maybe need to get word out to other people who can deal with the problem. Anyway, it turns out it was not a big league, sophisticated attack. It was something we've talked about, and in fact I've been talking about for the last couple weeks before this. Somebody opened an Excel file...

Leo: Oh, no.

Steve: ...containing a Flash movie.

Leo: Oh, no.

Steve: That's all it was. It was a so-called spear-phishing attack, meaning that - what RSA did reveal is that two small groups within RSA received some email that was targeted at them. So it was written to encourage them to open it. Well, it went, it was automatically routed into their junk email folders. So it wasn't even on their map. But one of the employees in one of these small groups looked in her junk mail folder, and the email was titled "2011 Recruitment Plan." And she opened the email, and there was an attachment, 2011 Recruitment Plan.XLS, making it a Microsoft Excel spreadsheet. That she opened, and that allowed a Flash movie, an Adobe Flash file that was embedded in the spreadsheet with an at-that-time unknown exploit, a zero-day flaw which Adobe has since patched, that allowed it to run. And that installed a well-known trojan which is freely available on the Internet called "Poison Ivy." It's a so-called RAT, an R-A-T, a Remote Administration/Access Tool/Toolkit trojan, which then phoned home, that is, it called outwards from her machine to a remote server that gave bad guys essentially the ability to do anything that she could do from her machine, they could do. And that's all it took. That was their foothold in RSA. And the rest, as they say, is history.

Leo: Geez.

Steve: So it was from, I mean, and I had been talking about Flash in embedded Excel spreadsheets for quite a while prior to this because this was a well-known problem that had been successfully used in spear-phishing attacks. It had been going on for several months. And unfortunately RSA was one of the victims of that. And that's all it took for someone to get enough access to their network that they were able to get credentials, elevate their access as necessary, and then exfiltrate what is now known to be essentially the keys to the kingdom, these master RSA files containing the mappings between the publicly available serial number of tokens and the secret key, the cryptographic key embedded in these hardware devices.

And as we know, at least one major agency has dropped RSA. I mean, they can't trust their two-factor authentication any longer. And RSA's advice was, well, don't let anybody see your token, and make sure all your other security is good.

Leo: Don't use us.

Steve: Yes, basically don't rely on that second factor. Oopsie.

Leo: Oy gevalt.

Steve: Yeah. Now, what's weird is that I had spoken to Stina Ehrensvrd, our founder of Yubico and great friend of the podcast about a month before at the, unfortunately or coincidentally, the RSA Security Conference. She was out here for that.

Leo: Is that put on by RSA?

Steve: Yeah.

Leo: Oh, wow.

Steve: Yeah. And, but, I mean, I'm sure funded and financed by lots of other people. But, I mean, RSA is like the main...

Leo: They have to be so embarrassed by this.

Steve: Yeah. It was not good.

Leo: They're supposed to be the security wizards.

Steve: Well, so there's a technology known as HSM, Hardware Security Modules. And these things are typically $15,000 just to get in the door. The idea is, if you've got your servers spread around, like in data centers, and they have really, really, really important stuff in them, how do you keep them physically safe and protected and network-level safe? That is, you know, as we've often heard, the main breaches are from internal employee subterfuge and sabotage of systems even to a greater degree than is external. We don't normally hear about it because they're not network-wide, sweeping, automated attacks. But much of the problem comes from employees. So how do you protect something from, essentially from yourself, but also from just, I mean, like if something is vitally important.

So this was the problem that Yubico faced as their own YubiKeys gained in popularity. They suffered an outage at one point earlier on, I mean, like not recently, but earlier on in their startup history that we're largely responsible for causing because we helped Stina get some traction in the industry. And that was a real problem for them, I mean, for their users because, in order to authenticate a one-time password device, you need some authentication server somewhere that's able to say, yup, that's the next password in sequence that we would expect.

So they realized they needed to, I mean, and they're still at this point a very small company, I mean, just a handful of people. So they're thinking, okay, we need to replicate this database of, basically of private keys, around the globe. We need - we can't just be in one location the way we are. It's time to get serious about this. But then they faced the question, how do we protect these? How do we have a server in Silicon Valley and one on the East Coast and a couple others in Europe? And the point being that all of these systems have to contain duplicate copies of everything so that local users are able to authenticate against them. But the flipside is, you have to keep them secure.

So they looked into the whole Hardware Security Module solution. And it turns out, I mean, again, it's $15,000 just to open the door. But you have to have, typically, you'd be repeating this everywhere. And you end up with, like, physical security and data level security. Well, what they decided was, Jakob, their lead techie person, said, why don't we put this in a USB stick? Like, put everything that we have to protect in a USB dongle? And let's build a cryptographic processor and a true random number generator, not just a pseudorandom number generator, but since we'll have hardware we can do weird things like measure the noise threshold in a diode which is actually generating quantum level noise and capture that and use that to generate cryptographic material.

And our listeners can't see it, but I am holding one up in front of the camera, which is - this is marked Beta 0.9.6. - is the YubiHSM, as they're calling it. It's the Yubico Hardware Security Module. They have essentially solved the problem and reduced the cost of protecting, literally, the keys to the kingdom, all the stuff that companies absolutely cannot lose control of, down from $15,000 to $500. So, and I imagine, I guess it does have - it's got their name on it, so they did some plastic molding. I mean, it looks, for the people who are not seeing the video stream, it looks like a regular USB thumb drive.

Leo: Yeah, same size, yeah.

Steve: That's all it is. It's got the USB interface. And so what they've built in here, there is a processor and nonvolatile memory and some fancy electronics that uses quantum level noise to generate true randomness, better than you can generate algorithmically. If our listeners are interested, just Yubico.com. If you look at, like, What's New, I think is the first link on their site, it'll take you to Yubico.com/yubihsm, where there's a lot of information. They're still at beta at this point. The final device will retail for $500.

And from their site they explain that it contains a cryptographically secure, true random number generator; a store for cryptographic keys, that is, a secure place to store cryptographic keys. It also incorporates a complete YubiKey authenticator that has enough storage for a thousand YubiKeys. So, for example, if you were a company that wanted to deploy YubiKeys yourself but not rely on Yubico for authentication, this little gizmo here can store your YubiKeys' secrets...

Leo: But you wouldn't - you'd need a server.

Steve: Well, yeah, exactly. So you plug this into a server, and so the server passes to this what the YubiKey has just said. And it says, yup, that's good; or, nope, that's not good. But the point is you are...

Leo: It's a black box. It's a sealed black box that can't be modified.

Steve: Precisely. It is a sealed black box that cannot be modified. And they list some use cases for this, saying you run an authentication service, secrets are stored on a computer that has to be accessible from the Internet and are concerned that one day it might be hacked. You want to prevent system administrators and staff who have physical access to the server from copying the database and getting access to sensitive data.

Leo: Yes, good point.

Steve: Yeah. You need an architecture that prevents a hacker from compromising your secrets, but allows you to run your service full speed. Or you've got a smaller fleet of YubiKeys and want to do the authentication yourself. So just today, on April 6, 2011, they have announced their YubiHSM. And they are making available betas if these to interested developers who want - because they're soliciting people to flesh out the full specification, to make sure they haven't forgotten something that might be useful to add.

Leo: That's the right way to do this, absolutely, get the security community to look at this.

Steve: Yes. So on their site it says, "Yubico is inviting its developer community to refine the YubiHSM and define the functionality set of the final product. Developers who would like to contribute with applications and further development of the open source client software can today apply to get a free beta YubiHSM from Yubico."

Leo: They anticipate, what, about $500 to buy this.

Steve: Yes, they're saying that it will be $500, which is way below the entry price, by orders of magnitude, what, 30 times cheaper than a $15,000 similar device which is the only other thing available now.

Leo: Now, somebody's saying that - a couple of people in the chatroom said, yeah, but look how easy that would be to lose. I don't imagine you'd keep that on your key ring. You'd keep it plugged into the server.

Steve: Oh, yeah, I mean, it's in a data center in a locked environment. But remember, if RSA had had these files similarly protected, they could not have been exfiltrated. And...

Leo: Now, even if somebody stole it, it's useless to them; right?

Steve: Correct, exactly, because it itself is protected cryptographically, and it cannot be induced through the API, there's nothing someone can do that can get the secrets out of it.

Leo: So even physical access to the server isn't going to compromise it.

Steve: Precisely.

Leo: But don't lose it.

Steve: No, well, no. I mean, you would have a couple of them. Or you would have this data backed up offline.

Leo: Oh, you could do that, okay.

Steve: The point is you don't want it online. It's like, I've never gone into the architecture of my own ecommerce system, but the data that we have is not available at that server. I did it from scratch the right way.

Leo: It's the equivalent of the cabbies' "Cash is not kept on premises."

Steve: Exactly. Exactly. And unfortunately, clearly, RSA's data was accessible for exfiltration. And this should be a lesson to all companies. And what I like about what Yubico has done is solving this problem no longer is necessarily super expensive. You could imagine smaller companies, developers listening to this podcast who are thinking, wow, right now we're just hoping we're not hacked. But if you move your stuff behind this kind of protection, then you don't have to hope any longer. You can know that you're safe against being hacked.

Leo: Awesome.

Steve: And they've got a PDF you can download that's got the entire specification of how this thing works, what it does, and how to talk to it. So I wanted to give everyone the heads-up that this thing now exists.

Leo: What a smart company.

Steve: Last week I deliberately didn't talk about something because it just seemed unlikely to me. I got a ton of email and tweets from people asking about Samsung's laptops coming pre-installed with keystroke loggers.

Leo: Little pat on the back for us. Because I saw that story, and I did not leap on it. And I know you didn't either. Something was fishy about that story.

Steve: Well, okay. The good news is our friend Alex Eckelberry - who used to be at Sunbelt but now is technically GFI's chief technology officer because GFI, I guess, bought Sunbelt - he stepped up with a blog posting that I really appreciated. The industry appreciated it because they explained what happened. This whole rumor of Samsung laptops coming with keystroke loggers preinstalled was the consequence of a false positive from one AV product, and it was Sunbelt's. It's their VIPRE, V-I-P-R-E. The Slovenian language directory...

Leo: Slovenian.

Steve: Slovenian, right, language directory for Windows Live was C:\Windows\SL. Which was the same directory created by the StarLogger, thus SL, StarLogger keystroke logger. And unfortunately somehow all that VIPRE was doing was looking for the presence of that subdirectory off of Windows to make its declaration. Now, Alex in his blog posted: "The detection was based off of a rarely used and aggressive VIPRE detection method, using folder paths as a heuristic. I want to emphasize 'rarely' as these types of detections are seldom used, and when they are, they're subject to an extensive peer review and QA process." So they recognized that, in retrospect, just looking for the presence of a subdirectory "SL" was really not specific enough, and it did...

Leo: No kidding.

Steve: Yeah. It would tend to produce false positives. In this case it did. And that sole thing was entirely responsible for the industry briefly going insane and calling and accusing Samsung of installing keystroke loggers when in fact that wasn't the case at all.

So for all the people who were concerned, that's essentially what happened.

Leo: Yay.

Steve: Yeah. Okay. Malicious code injection. Another event that's happened in the last week is the most successful SQL injection attack ever seen. Initially it looked like it was being reported as at least hundreds of thousands of websites. But later a Google search for domain names known to be used by the attackers uncovered more than three million pages, web links, which were displaying URLs to those domains. It was, not WebMon, shoot, I can't remember the name of the company that first uncovered, they named this LizaMoon because that was the first domain they saw - Websense, it was Websense...

Leo: Websense, that's right, yeah.

Steve: ...that ran across LizaMoon. And so they dubbed this the LizaMoon attack. But since then, 21 other domains were found. So what happened was someone created an automated SQL injector which rummaged through all kinds of websites looking for an SQL injection vulnerability.

Leo: And, by the way, somebody's saying in the chatroom this is not a new vulnerability they were taking advantage of.

Steve: Right, right.

Leo: It was an old one.

Steve: It was automated. And that's what made the difference. So all that had to happen was that, in your typical Web 2.0-style attack, a web server does not adequately sanitize user submissions.

Leo: It's supposed to, but that's just a mistake. It's a bug.

Steve: It's a bug. So basically a spider, a web spider posted these SQL commands such that, when the server went to display the page containing the content that the spider had put up there, instead it created links to these 21 separate domains which presented a dialogue box saying - a popup, basically, calling itself Windows Stability Center, looking like it was from Microsoft, though Microsoft doesn't have anything called Windows Stability Center. And so these were small businesses, community groups, sports teams, sort of mid-tier organizations who were hosted on any number of providers.

Leo: Everybody uses MySQL. Everybody. I mean, we use it.

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

Leo: You're smart.

Steve: ...everybody else does.

Leo: I wish we didn't, but...

Steve: Yeah, it's just you have to be extremely careful when this is the kind of technology that you use, that you've got filters.

Leo: To illustrate that this is a longstanding problem, there's a great comic I'm going to show here from XKCD, our chatroom reminded us, XKCD.com/327. "Hi, this is your son's school. We're having some computer trouble." "Oh, dear - did he break something?" "In a way. Did you really name your son Robert'); DROP TABLE Students;--?" "Oh, yes. Little Bobby Tables, we call him." "Well, we've lost this year's student records. I hope you're happy." "And I hope you've learned to sanitize your database inputs." That is a very geeky comic, but boy, it says it right there. That says it all.

Steve: Yup, exactly.

Leo: That's obviously an SQL command that says delete a table, the entire students table. And if you didn't check your inputs, if you allowed somebody to type that into a URL, well, shame on you.

Steve: Right, because then, when the server tried to display it, it would execute it as if it were a valid command.

Leo: Oh, lord.

Steve: And your SQL database would drop that table.

Leo: Say, yeah, whatever you want. The problem with computers, they're very compliant. Okay. Let's talk Epsilon.

Steve: Okay. So Epsilon is the world's largest permission-based email marketing services company.

Leo: Who knew?

Steve: I know, never heard of them before.

Leo: Who knew, yeah.

Steve: 2,500 clients, including seven of the Fortune 10. So seven of the largest 10 corporations in the United States use Epsilon to do their customer emailings. So when we get email from, well, from 1-800-Flowers, AbeBooks, Air Miles, Ameriprise Financial, Barclays Bank, Beachbody, bebe stores, Best Buy, Brookstone, Capitol One, City Market, Citi, Dillons, Disney...

Leo: And he's just in the C's.

Steve: ...Destinations, Eileen Fisher, Ethan Allen, Food 4 Less, Fred Meyer, Fry's, Hilton Honors Program, Home Shopping Network, Jay C, JPMorgan Chase, King Soopers, Kroger, Lacoste, LL Bean Visa Card, Marriott Rewards, McKinsey & Company, MoneyGram, New York & Company, QFC, Ralphs, Red Roof Inn, Ritz-Carlton Rewards, Robert Half, Target, The College Board, TD Ameritrade, TiVo, US Bank, and Walgreens. They were all breached. That is, Epsilon is saying that 2 percent of its email clients, which 2 percent of 2,500 would be, what, 50, were affected. Well, I just read the list of known clients who are now vulnerable to a much heightened level of spear-phishing. The problem is that what was lost was the email databases for those companies.

Leo: So people didn't get - they didn't get the credit card numbers or personal information. They just got email addresses.

Steve: Well, and names.

Leo: And names.

Steve: And that's the problem is that there's now - there's a much greater chance that you will click on a Hilton Honors Program email that knows your name.

Leo: Hi, Leo. Your Hilton Honors program is about to expire. Want to renew, make sure those miles are still good? Click this link.

Steve: Right.

Leo: And then you log in.

Steve: Right.

Leo: But it's not them.

Steve: Right. So, I mean, because of the breadth of this, I mean, this was a huge breach of their database. And once again one wonders, if they had protected this in a Hardware Security Module of some kind, making it not available for theft, then they wouldn't have egg on their face. So other security blogs and security-aware people are really, are warning people to be especially alert for spear-phishing attacks. That is, from the list that I read, which is the most complete list I've managed - I pulled it from...

Leo: Is that from the Times or...

Steve: ...a bunch of different places.

Leo: Oh, different places, okay.

Steve: Yeah. And that's why I had to alphabetize it in order to, like, make sure I hadn't lost any and to remove duplicates because there were so many companies there. But, you know, their email addresses are - their email lists are out, along with the people's names. That is, whatever it is that you are - what you normally see when you receive email from these companies is it comes from the database that was lost. So the point is, that's what you'll see when you get fraudulent email and may be much more inclined to click on a link. And so it's dangerous.

Leo: And the funny thing is that Epsilon gave boilerplate language to send out. All these companies sent out essentially the same email. And unfortunately, now, Chase's was pretty good. Somebody sent me the Chase email, and it says, "We want to remind you, Chase will never ask for your personal information or login credentials in an email. As always, be cautious if you receive emails asking for personal information," et cetera. Many of these - actually the Chase email was pretty good. It was pretty good. It actually said don't click links and stuff. Most of these, the one that Epsilon seems to have sent out to everybody, said don't click links on email from people you don't know. But that's the problem. It's going to come from these companies.

Steve: Right.

Leo: It's not going to be from people you don't know, it's going to be from Citibank and Chase.

Steve: Exactly. Essentially, the data that was lost, that they lost control of, that was exfiltrated from them, is what they use, is exactly what they use to generate the legitimate email because they're the people that send the email out on behalf of their clients.

Leo: Right. So what is the advice? I mean, I know our audience knows it. But just to reiterate.

Steve: Well, I would say, if you receive email, because email itself, email displaying in a web page where the URL behind the link that you click can be masked...

Leo: The link can say Citibank.com, click here, and go to Hacker.com/Citibank.

Steve: Or C-i-t-y-b-a-n-k, I mean, a tiny change in the domain name takes you to somewhere else. So, I mean, unfortunately it's really - I mean, again, this is what our listeners know. It is just not safe to click on a link in email, that you receive in email. If you really have to, for some reason, you could look at the email headers, maybe. But unfortunately you've got to be a real expert because you and I have talked long ago on TechTV shows, Leo, about spoofing email headers and how easily that could be done.

Leo: It's trivial.

Steve: So, I mean, the real news is read the email, then manually go to the website, entering the URL yourself, logging in, not through email, but using LastPass or whatever you use for logging in, and arrange to achieve the same end, but not clicking something that you receive in email. Treat the email as just the information that something is important that they're bringing to your attention, like, oh, look, your miles are about to expire unless they hear from you immediately.

Generally the phishing emails use an emergency of some sort to get people to act. They're not saying, hi, we just wanted to make sure you're happy with the service we're providing you because people go, yeah, yeah, yeah, fine, delete. No, it's that there's a call to action in spear-phishing emails that is presenting you with some dire event unless you take action. And people go, ooh. And in the moment of worry, they hit that, they click on that link without - even if they know better, it's like, oh, I'd better do this right now. It's like...

Leo: I've come so close because they scare you. You click the link, and it all looks legitimate. And then fortunately in my case I've always gone, whoa, whoa, whoa, as soon as it asks you for anything, and you look at the URL. But you're right, hand-type the - that's what I told everybody on the radio show. So thank you for reiterating that. That's all you have to do. Hand-type it.

Steve: Yes.

Leo: Now, there is one - and the chatroom came up with this. There's one side benefit to the fact that Epsilon services so many of these big companies. You can go to their website and opt out of all emails with one click. So if you go to Epsilon.com or search for Epsilon consumer opt-out information, there is a page where you can say don't send me anymore crap.

Steve: Well, and are those only promotional emails, or are those, like, true account maintenance sort of emails?

Leo: Well, it's marketing emails.

Steve: Oh, okay.

Leo: So it says "Many consumers value and seek out targeted advertising [bullshit] and look forward to receiving offers of interest at their homes via email." So you can choose...

Steve: That's amazing if you can opt out of 2,500...

Leo: You can choose not to receive most - it doesn't say all, but most targeted advertising by following the links below. Consider your choices carefully. Opting out of Epsilon Services will stop the delivery of some targeted advertising.

Steve: Some spam is good. Uh, right.

Leo: It will not eliminate all targeted offers, it says. So much for that. But at least some. I've been, lately, I've been unsubscribing. I've been on an unsubscribe tear. Anytime I get something, any bacon, I just unsubscribe, unsubscribe. And I'm hoping that that will take hold.

Steve: So the do-not-track movement is gaining some traction. For people who use Twitter, there is now an account, it's just @donottrack. And I am following it and would recommend it for anyone who's interested. Mozilla blogged at the end of March, since we've last spoken to our listeners through this podcast, that advertisers and publishers are beginning to adopt and implement do-not-track. Specifically, in their blog posting, they said that the AP news registry service run by the Associated Press implemented the DNT header across 800 news sites, servicing 175 million unique visitors each month.

And the Digital Advertising Alliance, which actually I care about much more, the DAA, which includes the five major media and advertising agencies, is initiating a process to explore incorporating the DNT header, as proposed by Mozilla, into its self-regulatory program for online behavior advertising. The DAA represents more than 5,000 leading media and technology companies that span the entire marketing media ecosystem. So the good news is, this is the same header that IE9 offers.

Leo: Oh, good.

Steve: That DNT: 1. And I immediately contacted Giorgio, the famous author of NoScript, whose headers were not compatible; and I said, "Hey, looks like this has pretty much been decided. How about switching NoScript over?" And he said, "Steve, I did that a couple months ago."

Leo: What a surprise.

Steve: Ah, oh. Okay, good. So the good news is, if you - and of course Giorgio reminded me also that using NoScript to produce the header allows you to allow some and disallow others. So it gives you granular control. Whereas the other guys are just a blanket, stick this DNT: 1 header into every request made by the browser. Which I think is just fine.

So if you haven't yet gone up to v4 of Firefox - and I have not because I don't do anything immediately when this sort of thing happens, I wait a while - but you are using NoScript, NoScript is now adding this DNT header, as is IE9. You have to turn it on in every case. Nowhere is it enabled by default. So you do need to opt into opting out. But at least it's there. And everyone, all the critics say, yes, well, but it's all - honoring it is optional. It's like, yes, it is at the moment. But, you know, these advertising agencies recognize that they risk having Uncle Sam in the U.S. case drop a heavy foot on them unless they behave well.

Leo: Yes, exactly, yeah.

Steve: And so this is - ultimately we're going to get a law that says, if your browser says DNT: 1, it is illegal to track you across the Internet. And we're not there yet, but that's the right answer. And it'll end up happening.

Leo: And even if we don't get a law, the real problem is bad guys will always circumvent, no matter what. But the good guys are going to try, look, they want to stay in - people like Epsilon want to stay in business. They're not going to flout, flaunt our requests.

Steve: Right, right.

Leo: I hope.

Steve: Right. A clear and explicit request not to be tracked behaviorally.

Leo: The problem is I just was following the rabbit hole of this Epsilon opt-out.

Steve: Uh-huh.

Leo: They don't make it easy. They ask for - and it's not optional, required - the header from the spam, from the message that you - not who it's from, not - they want the entire header. Most people will not know how to do that and will not do it.

Steve: Right.

Leo: Terrible. Just awful.

Steve: So I've got a little bit of miscellaneous stuff before we get into talking about Comodo. This just crossed my radar, and I thought this was really interesting: Microsoft offered to purchase 666,624 IPv4 addresses...

Leo: We're running out.

Steve: ...out from the bankruptcy proceedings of Nortel, the bankrupt Canadian telecom equipment maker. So Nortel's going bankrupt. They're a corporation that had a huge block of IP addresses. And Microsoft has offered and apparently - I just saw something else that looked like it had been accepted - $7.5 million for their block. And that's, like, wait a minute. Since when are IPv4 addresses for sale? I mean, you know, you get them for free when you have an ISP. So I'm interested in how that can be regarded as a corporate asset, why they wouldn't immediately return to the provider of Nortel's - wherever Nortel got them from, from some registry. It's like, wait, they're not property, they're sort of - I don't know what they are. But I don't know how you can buy them.

But Microsoft's paying $11.25 each for these. And as you said, Leo, it's like, oh, no. We're running out of them. And I have seen other notes talking about the emergence of a black market, or a gray market, at least, or dark gray, in IPv4 addresses. I guess they're going to become worth something. Wow. I don't know how you can sell them. I don't know you...

Leo: You could auction them off.

Steve: ...can call them property.

Leo: But you know, it does change bankruptcy proceedings. You now actually have some assets that might be of some value.

Steve: Yeah.

Leo: Ignore that person looking over my shoulder, it's my son. Go ahead, Steve.

Steve: Someone tweeted something that I really appreciated to me, that I wanted to share. Just noting, he said, "Steve, you don't need to be quite so embarrassed. Skype's SSL certificate expired on March 31st."

Leo: Well, it's about time.

Steve: And of course caught them with their pants down.

Leo: Isn't that funny?

Steve: It's happened to me; it's happened a couple other times to notable large organizations. And so you go, oh, crap, and you immediately go with your registrar and renew your SSL certificate and get it up on your servers. So it does happen, even to the big guys.

Leo: Amazing.

Steve: I also wanted to - just a note that there is a forthcoming virtual machine solution to allow Android to run on Windows, called BlueStacks.com. BlueStacks...

Leo: That's cool.

Steve: Yes. It's not out yet. They ask you to follow them on Twitter @bluestacksinc, so it's just @bluestacksinc. If you go to BlueStacks.com you get a little Flash video that plays and sort of shows you something, some pretty graphics talking about the idea of being able to have Android apps running in a VM on Windows. So we talked about how Android was hosting a VM last week and would allow you to use that in order to preview Android apps. Now you'll be able to do it locally, running them under Windows. There is also something called Android-x86.org, which apparently has made this happen also. But I'm told that it's much more difficult to get it up and configured and with the video drivers and all kinds of weird stuff. So with any luck, a VM would be a great little solution for basically giving you a little Android machine running under Windows that you could run Android apps in.

Leo: I can't wait to do this. I wish it were there now, actually.

Steve: Yeah, we'll keep an eye on it. And I am following bluestacksinc in one of my following Twitter accounts, so I'll...

Leo: Yes, I see them. I just followed them myself because I...

Steve: Yeah, but you follow a thousand...

Leo: I know.

Steve: I mean, 17,000 or something. So you'll never know if it happens. But I'll let you know. I'm sure you'll know before I know, Leo.

Leo: I'll let you know.

Steve: That'd be very nice. And I wanted to mention I got a nice little note from a Neil Warwick with a subject that caught my eye, saying that "SpinRite Saves the Sky." And I thought, okay, what? Okay, so and he's at Reading, England. He's a Security Now! listener. He sent it through the Security Now! feedback form at GRC, which is GRC.com/feedback. He said, "Hi, Steve. Just thought you'd like to hear a short story about how SpinRite saved my Sky TV/DVR from losing all my partner's saved shows, and probably saved me from a lot of nagging, too.

"In the U.K. we have a satellite TV service called Sky TV that broadcasts many, many channels. And for most people it's accessed using a hard disk-based DVR similar to your TiVo boxes. A few weeks ago our box started freezing and stuttering when viewing live or recorded content. We have an insurance package with Sky to cover breakdown, so I called them. Their only solution was to offer to send an engineer out who would reset, in other words, reformat the hard disk; then, if that didn't work, would replace the DVR in its entirety. Having been a listener of Security Now! since July 2010, when I saw it mentioned in a magazine" - I don't know if he saw SpinRite or Security Now! mentioned - he says, "I started at the beginning." I think he saw Security Now! mentioned in a magazine. So in July 2010 he saw Security Now! mentioned in a magazine.

He says, "I started at the beginning; so as I write this, I'm up to about Episode 155. I've been looking for an excuse to buy a copy of SpinRite and thought, why not give it a go? 15 minutes later SpinRite was downloaded and burned to a CD. I removed the hard drive from the DVR, installed it into a slave PC chassis I had lying around, and booted the CD. I set SpinRite to work immediately and went to bed.

"Upon getting up the next morning, SpinRite had finished and reportedly had fixed some errors. So I reinstalled the drive into the DVR and powered up. As you will probably have guessed by now, everything worked perfectly. When the engineer showed up to have a look, he couldn't find anything wrong with the system so left without doing anything. I didn't tell him about SpinRite, as I'm not sure if I'm allowed to open the box under my insurance agreement. Thanks again for a great product and great podcast. Neil Warwick."

Leo: Awesome, awesome, awesome.

Steve: It says, "P.S.: I run a very small computer repair company and would like to offer a SpinRite optimization-slash-check as a service for clients' PCs, et cetera."

Leo: That's a good idea.

Steve: "Can you tell me what kind of license I would require to do this, and how much it would cost?" Well, the way we've solved that problem is to ask people who want to run SpinRite on machines they don't themselves own, or a corporation that just wants to run it across their whole organization, we call it a site license. And so we ask people to keep four licenses current. So, for example, Neil already has one. So if he bought three more copies, then he would qualify to use SpinRite on any drive that he wants to. That's essentially a site license.

Leo: That's a great deal. That's a great deal. I think that's very fair.

Steve: Yeah, well, it allows people to try it and then not - then they don't have to ask for a refund when they want to upgrade to a different license. So I thought, okay, let's just have them maintain four current licenses. So when we have an upgrade, if they upgrade all four, then that's like upgrading their whole license. So but the plan works very nicely.

Leo: So let's talk about this Comodo breach.

Steve: Okay. So here's - there's a really interesting back story here because it wasn't initially acknowledged by anyone. What happened was a researcher at the University of Washington's Security and Privacy Research Lab, Jacob Appelbaum, he was just sort of, as he does, monitoring the Chromium, that is, Google's Chrome browser open source project, just sort of - he's like part of the change log, seeing things go by. And in late March he noticed sort of an odd thing had been added. A bunch of SSL certificate security serial numbers had been put into the source code and blacklisted. So there was a function that had been added that said, essentially, see if certificate is blacklisted.

And then there was an array of serial numbers. The first one was commented as this is just to be used for testing. But then there was, like, nine or 10 others after it. And he thought, huh. That's weird. Why would Chromium be, like, putting a block of SSL certificate serial numbers into the source? Seems like a strange thing. And this sort of just, like, raised his curiosity.

And then, at around the same time, he noted that Mozilla pushed out a security update. And he looked into it, and it was doing the same thing. It had some - he was a little bit thrown off at first because Mozilla's format has a zero byte prepended to the serial numbers. So a simple match didn't line up. But when he removed that leading zero, then he realized that Mozilla was also blacklisting, in a patch, a block of SSL certificates.

Well, he hangs out at EFF, and EFF has something they call the SSL Observatory, where they monitor the Internet, looking sort of at traffic, sort of just trying to watch the way SSL is being used and seeing what's going on. The SSL Observatory had built up over time a big database of certificates. So he was able to essentially process this list and see that it appeared that all of these certificates had been issued by one certificate authority, something called UserTrust.com. The URL in the certificate was http://www.usertrust.com. And that seemed to be a subsidiary of some sort of Comodo.

And so he sort of thought, okay. It sure seems to me like the only reason Chrome and Mozilla would suddenly be blacklisting a block of certificates is that they're important, and they're bogus, they're invalid. So he sent a note to Mozilla, who said, uh, hmm, yeah. Something's happened that we don't want to talk about. Now, in fairness to Mozilla, they have since regretted their response, that they weren't immediately forthcoming.

But what happened was there was some dialogue behind the scenes. What essentially happened, we now know, is that a subsidiary of Comodo - and Comodo has never even said whom. But we now know, from this forensic analysis, one of Comodo's - Comodo is a so-called CA, a Certificate Authority. They have a subsidiary that they call RAs, Registration Authorities. And one of their RAs was hacked in some fashion, and we think we know how now, and a bunch of very high-profile websites got loose, well, high-profile certificates for websites were made illegitimately: mail.google.com, so an SSL, a valid SSL certificate for mail.google.com, for www.google.com, for login.yahoo.com, for login.skype.com, for addons.mozilla.com, for login.live.come, which of course is a Microsoft domain, and something called "global trustee." Valid certificates for those high-profile domains were issued by this registration authority that has received its authority from Comodo.

Now, the SSL Observatory, this project that the EFF runs, was able to state that, knowing that these certs were signed by this user trust organization, as of August 2010, so late last summer, 85,440 publicly transacted, that is, they've seen them on the Internet, HTTPS certificates were signed directly by this UserTrust organization. That's significant because, if we were to blacklist UserTrust, then 85,440 websites would suddenly have their certificates that they had gotten from this UserTrust organization declared invalid, and they'd have to scramble around to obtain SSL certificates from someone else.

So after this became public - oh, the other thing that happened was that I first became aware of it when Microsoft issued an emergency update through their Windows Update system. And looking at it closely, it was very clear that Microsoft was immediately adding a block of certificates to their untrusted list in Windows. I tweeted immediately to the followers of SGgrc that this had just happened, and gave them the link to Microsoft's security page, where you could choose which operating system you were using, and yourself immediately update your version of Windows so that it would no longer trust these certificates. And the next day it surfaced, or shortly thereafter surfaced on the standard Windows Update as an important update that users should install immediately. So Microsoft, Google's Chrome, and Mozilla were all immediately updated.

Now, one of the questions this raises is, well, okay, don't we have a facility in place for revoking bad certificates? And the answer is, uh, kinda. It doesn't really work. And that's why all of these major browsers, well, these three major browsers were immediately updated. But what about less significant browsers? What about browsers that are not mainstream? There's lots of little offshoots that didn't get changed, didn't get updated, that aren't part of the central core browsers. To what degree are they vulnerable, and to what degree are their users vulnerable?

So the other thing that happened when news of this surfaced is that someone began posting on Pastebin. We've talked about Pastebin once before because there was - it's a way of anonymously posting stuff up to the Internet that anyone who knows the URL - then you share the URL, and people can go there and grab what you posted. A hacker was claiming that he's the person who did this and put a bunch of code and certificates and things up in order to prove it. Several people have in fact verified, because he posted the private key which he used, and that allowed people to verify that he was in fact the person who created these certificates. What apparently happened was that this registration authority, who is a subsidiary of Comodo, had a DLL which itself was empowered to log onto their servers and issue certificates. So this guy, who is believed to be an Iranian hacker...

Leo: Oh, great.

Steve: There was some news, there was some speculation that this was state sponsored, that this was based in, like, Iran was hacking SSL in order to get certificates that could be used to spy on people. It's possible; but it's, again, we generally like the most feasible explanation. And it looked like it was one guy because, if this was state sponsored, no one in Iran would have gone bragging about this and posting this stuff on Pastebin...

Leo: Well, that's true, yeah.

Steve: ...and so forth. So he reverse-engineered this DLL which contained the username, the login username and password for the server that this subsidiary of Comodo, apparently UserTrust, was using, and accessed their server and was able to induce it to issue these certificates. Now, Comodo found out about it, immediately revoked these certificates. We'll talk about that quickly. But then they also did find one server in Iran where one of the login.yahoo.com certificates was in use, meaning that, like in the same way that when Google has, Google gets a certificate for themselves, they install it on their servers so that when you connect to them over SSL, that server is declaring, I am Google.com. And that's the whole point of a certificate is that it is providing authentication in addition to encryption. So it was briefly the case that a legitimate server in Iran, at an IP located in Iran, was saying it had one of these fraudulently issued certificates installed on it so that, when you connected to it in Iran, it said, I am the login.yahoo.com server. And it quickly stopped answering queries after this revocation happened.

So, briefly, to talk about revocation. The idea is that certificates have a certain life. I like them, you know, I complain about having to do this every couple years. But there's an upside to it because, by keeping certificates relatively short-lived, they will expire no matter what. And so the burden is on the certificate licensee, the owner, to go and renew the certificate with the certificate authority and get another one that's good for one, two, or three years. I think three is the most I can purchase from VeriSign, which is where I get mine. But what happens if, while during that window of time that a certificate is valid, if something happens, for example, if the certificate got away from its owner?

So, for example, if Google lost control of the private key that is what makes its certificate valid, which has been signed by VeriSign, if they lost their certificate, then, that is, if anyone else could get it and install it on some random server, that would be bad. So if that happened, Google could say to whoever signed their certificate, please revoke this certificate. We need a new one, and we need the one that we lost control of to be canceled.

So there's a facility - there's actually two. One's called a CRL, a Certificate Revocation List; and the other is a protocol, an Online Certificate Status Protocol, OCSP. The certificate revocation list is a list which a browser client can download from the Internet containing a long list of all the certificates which are revoked, but otherwise valid, meaning that, thank goodness for the expiration of certificates, so this list doesn't have to contain certificates which are expired. It only needs to contain them, the certificate serial numbers of certificates which would otherwise still be valid except that they have been revoked by their issuing agency. And the point is it's only by somehow proactively declaring a certificate invalid that it can be found to be invalid. That is, if bad guys get it, then they're going to want to use it.

So somehow we need to tell our browsers, no longer trust this certificate; however, still trust other certificates issued by the same certificate authority. Now, and of course we've talked about the controversy, the fact that there's more than 1,500 certificate authorities wandering around the globe now, all equally trusted by our browsers. Okay, the second mechanism, this online certificate status protocol, that's a different approach where, before trusting a connection, the browser will reach out and, in real-time, check to see that a certificate is valid. So that's something that some browsers support that you can turn on to check it on the fly.

The problem is, both of these approaches fail open, meaning, if your browser is unable to obtain information that the certificate is bad, it assumes it's good. So part of a spoofed attack, a spoofing attack would be, for example, a man-in-the-middle attack, where somebody would intercept your communication - we've talked about this in WiFi scenarios or hotel scenarios or state-level scenarios. They would intercept your connection, and they would proxy your access to the Internet. They would, when they saw you trying to go to login.yahoo.com, they would accept the connection, respond using their fraudulent and revoked certificate, which your browser doesn't know is revoked. When your browser tries to authenticate on the fly that revocation, all they have to do is drop the connection. All they have to do is return a 404 or a 500 server error, or bad URL, something that blocks your browser from getting an affirmation or denial, and all browsers currently accept the certificate. Even though technically it's been revoked.

This is regarded as a known and serious problem with the whole revocation process, is that the same mechanism that would have a bad guy able to pull off a useful exploit with the certificate, unless they were able to spoof DNS, which is the other way you'd do that, is to get someone to go to a bad IP address by spoofing DNS. But any kind of a man-in-the-middle approach would still have your browser believing that the certificate was valid because it wouldn't know otherwise.

And when we step back from all this, here's the fundamental problem. When you think about the way our whole SSL browser security web authentication system functions, the fundamental technical design is fragile because any certificate authority can certify to any user that any server owns any domain name. Therefore, the consequences of a misplaced trust decision are about as bad as they could be. And stated another way, the problem is a structural one. There are 1,500 certificate authority certificates, controlled by about 650 organizations. And every time we connect to an HTTPS web server or exchange email encrypted by TLS, we implicitly trust every one of those certificate authorities because, think about it, we've got these, as we've talked on this podcast, a huge block of certificate authorities we trust. When we receive a certificate claiming the identity of a server we want to connect to, it is signed by one, any one, of those. Meaning that any one of those can sign a certificate, and we will trust it. So we are trusting all of them.

Leo: Okay.

Steve: Yeah.

Leo: Wrong way to do it, obviously.

Steve: I read, during the research for this, one interesting concept that I just - I haven't even had a chance to think through enough. But the concept is a different way of structuring the system where the people having the certificates are - it turns things around. Instead of having a multiyear expiration, you have a multiday expiration. All of your web server certificates expire every three days. And then you - so it's necessary for you - and this obviously would be automated in some hopefully good fashion - it's necessary for you to go out and renew, you know, you have your web server establish a trust relationship with a certificate authority and dynamically renew your certificate every other day so that you're sure you're going to get it renewed before the one you have expires. Or renew it every day or something.

So but it's sort of a - again, I haven't had a chance to think that all the way through. So, but it's sort of, again, I haven't had a chance to think that all the way through. That's an interesting, completely different approach to building this trust. Fundamentally the problem is we trust too many people. I mean, trust is too widespread. And there was a blog posting asking, well, do I really need to trust a certificate authority in China?

Leo: No.

Steve: And I really don't think I do. I mean, I'm not going to Chinese websites because I can't read them anyway. So why do I have that certificate authority in my browser which is subjecting me to the danger of going to a spoofed website using a domain, a super popular domain? Oh, and that brings up one other issue. There was a bunch of back flak aimed at Comodo, as you can imagine. And one of the real good criticisms was, wait a minute, you know, not all web domains are created equal. Arguably, Live.com, Google.com, Yahoo.com, Microsoft.com, I mean, there are, in any statistical distribution, there's a relatively small set of super high-value domains, I mean, major focuses of the Internet. And you would really think that all certificate authorities should be, like, have a database of, absolutely, bells and whistles should go off, sirens should sound, if anyone tries to register any of this set of really high-profile domains. Why was it that some second-tier certificate authority even had the ability to issue certificates for www.google.com? And, I mean, it really does say that this infrastructure could be hugely strengthened.

Leo: Well, I'm glad you talked about it. I can see why they haven't changed things. You change something like that, it breaks a lot of things.

Steve: Oh, my goodness, yes.

Leo: And who do you call for support? So I can see why there's a lot of pressure not to change things. But, boy, that's a mess.

Steve: Actually, though, it is the case that just changing that certificate expiration model, nothing else breaks if you change that. I have to think about that some more.

Leo: Yeah, let's - yeah. Well, I'll tell you what, next week, feedback episode.

Steve: Yes.

Leo: Guess what? We're going to probably hear from some people who have some thoughts about that. Maybe you get to vet Steve's idea, and maybe we'll get an A+. GRC.com/feedback is the place to go if you have a question for Steve. We'll answer as many as we can on the next show. Every other show we do that. While you're there, get that copy of SpinRite you've been holding out on. It's well worth it. If you have hard drives, you need SpinRite, just the best hard drive maintenance utility. Nothing better. GRC, Gibson Research Corporation, GRC.com. You'll also find a lot of free stuff there, including 16KB versions of this show, transcripts of this show, full show notes going back 295 episodes, it's all there at GRC.com.

Steve, as he mentioned, is on Twitter at SGgrc. He's also got a corporate account, @GibsonResearch. I don't know if you do anything more on the Pads, but he's got @SGpad, as well. Follow them all. Follow all three. Why not? And, Steve, I apologize for the interruption from my son. And we will find the missing clock.

Steve: That was fun. I will talk to you next week for a Q&A.

Leo: Thank you, Steve Gibson. We'll see you next time on Security Now!.