Listener Feedback #82

Description: Steve and Leo discuss the week's major security events and discuss questions and comments from listeners of previous episodes. They tie up loose ends, explore a wide range of topics that are too small to fill their own episode, clarify any confusion from previous installments, and present real world 'application notes' for any of the security technologies and issues we have previously discussed.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-228.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-228-lq.mp3

Leo Laporte: This is Security Now! with Steve Gibson, Episode 228 for December 24, 2009: Your questions, Steve's answers #82.

It's time for Security Now!, the show that covers everything having to do with your security, your privacy, your safety online. And here he is, the secure, private, safe Steve Gibson. Hello, Steve.

Steve Gibson: [Laughing] Hey, Leo.

Leo: How are you today?

Steve: Great to speak with you again. I'm great.

Leo: Our holiday edition.

Steve: Yes. And based on the weather on the East Coast, I'm glad we're both on the West Coast.

Leo: No kidding, no kidding. It's Christmas Eve. And you are visiting your mother.

Steve: Yup, well, the whole family. I grew up in Northern California and wandered down here following my strange career path. And Mom used to, you know, back then we didn't

have the Internet so she'd - we'd talk on the phone. She'd say, "Honey, when are you going to come home?"

Leo: Aw.

Steve: And I said, "Mom, you know, this is just the right distance."

Leo: [Laughing] I know.

Steve: If the faucet's leaking, I'm sorry, I can't help you with that washer. Go get my brother-in-law. He'll grumble a lot, but do it.

Leo: I'm safe on that count because Mom lived in Santa Cruz and then moved to Rhode Island. I stayed here. So she moved away from me. But she lives two blocks from my sister, so...

Steve: Yeah. And I like it that it's just a short little hop up the freeway. I mean, not the freeway, just a short little hop up the - by plane it's like an hour. And so it's close enough that it's easy and not a problem. So, yeah, it works just great.

Leo: Well, we are gathered today to answer listener feedback. This is our 82nd listener feedback episode.

Steve: Yup.

Leo: We're back on the even number, the mod-2 episodes for our Q&A. And I bet you we have some questions about last week's episode on cyberwarfare.

Steve: We've got some interesting things there, of course. One person gave me the perfect opening to talk about this ridiculous, the unmanned Predator unencrypted video mistake that the Defense Department made, so we'll talk about that. Lots of interesting security news. And I managed, believe it or not, I still managed to somehow find a SpinRite testimonial unlike any others. So...

Leo: You'd think we have heard them all.

Steve: You would have thought we would have done them all, but we've got a new one. So, yup.

Leo: Well, let's start with any security news or errata we want to correct?

Steve: Yup. Well, once again, Adobe is in the doghouse.

Leo: Uh-oh.

Steve: There is a publicly exploited, confirmed by Adobe, critical remote code execution vulnerability...

Leo: Again.

Steve: ...again, in Reader and Acrobat, any versions current and prior. So right now they're at 9.2 of both Adobe Reader and Adobe Acrobat. And those and everything before are vulnerable. The nature of the problem is a so-called "use after free" error. We've sort of talked about that. The idea is that many systems dynamically allocate memory as needed. So the code'll be going along somewhere, and a packet will come in, and the program doesn't have anyplace to put it. So the program asks the operating system, this is one of the functions of an operating system is to provide memory on demand when asked for by code. So the code'll say, give me a chunk of memory, and the OS will say, okay, here's a pointer to the chunk of the size you asked for. The program will then do whatever it does with it and use it for a while and then free it. The idea is that it dynamically asks for it as it needs it. And when it's through with it, it releases it.

And long-time listeners of - long-time listeners. Well, long-time users will remember the old days of Windows where there were things called "memory leaks," where remember you'd, like, use Windows for, like a few hours, and you'd run out of memory. There was, like, it would crash. Well, what those old-style memory leaks actually were, they were programs that forgot to release the memory that they had asked for. And so a program would just keep asking for memory, intending to let it go, to give it back to the operating system so that it could be reused. But there was a bug in the program, and it would so-called leak memory, meaning that it ended up just consuming it but never releasing it. So in this case...

Leo: And in most cases modern memory managers and garbage collection and programming languages have taken care of that.

Steve: Well, yes. Although in some ways that don't require the programmer to behave himself. For example...

Leo: Encouraging lazy programming, yeah.

Steve: Right. For example, what'll happen in - as soon as we moved from the 16-bit world, Windows 3.1 and 95, 98, and ME, there was a chance to change the paradigm. And starting with Windows NT, when a program terminated, the OS had the ability to robustly release any unreleased memory. So frankly, the problem went away, not because all the programmers really got good about behaving themselves, but because the operating system started taking responsibility for their misbehavior. An example where you can't do that is a server app. Like all of the code that I write I don't - I haven't rebooted my server in years.

Leo: Right. No memory leaks there.

Steve: No memory leaks. And, I mean, there have been problems where - in fact there was one a few months ago, toward the end of the work on the spoofability test. It turns out that I was creating threads, and there were some thread handles that I was not freeing. And so over the course of weeks I would see the server's consumption slowly drifting upwards. And it's like, ehhh, you know, I can't live with that. So I just put some time aside and reread my code. And I said, yeah, you know, I don't think I'm releasing the handle of those threads. And sure enough, added that one little line, and then the problem went away. And I'm back to absolutely robust long-term operation with no memory leak.

So what happens is that, if programmers are not very careful, it's possible to re-use a pointer to memory which has been freed. And that's the so-called "use after free" error. In this case, in the Adobe products, there's a method in the code called newPlayer. And there's an object called the Doc.media object. And by exploiting the way it works, it's possible to allocate memory, put what you want in memory, that is, for the attacker to load their code in there. But due to the nature of the way it's set up, you're not supposed to be able to execute the code. And in fact you can't. But if you release it to the operating system, saying okay, I no longer need this, and then use the pointer, you're able to access the code, which you would otherwise be prevented from accessing. So it's a mistake which Adobe made.

And as I was looking at the details of this, and the fact that we talk about Acrobat or Adobe problems pretty much every single week now, I'm wondering what's this escalation? And I think it must be that hackers have realized there is a body, a large body - and if you've installed Reader lately you know how large the body is - of code which wasn't very well written. I mean, so they sort of painted a target on themselves. You know, for a long time they got away, while everyone was aiming at Microsoft and calling Microsoft a big, bad, security vulnerability, Adobe was busily adding features and bloating up their software, not paying attention to security. And so here now they've got - they've got code that they're late in the game in raising the security bar on. And the attackers have said, let's take a look at this Adobe stuff. Everybody else is finding problems; maybe we can. And, you know, wherever you look there's a problem.

Leo: It's a fertile field, isn't it.

Steve: Exactly that, yes. Speaking of which, Mozilla is back in the game. In this case, though, they do have a patch. Oh, I forgot to mention that Adobe has indicated that they're going to fix this with their planned January 12th update. And I'm not kidding, their advice is disable JavaScript.

Leo: As usual. As always.

Steve: As we've heard before. However, this is not the first time our listeners have been advised to disable JavaScript in Acrobat Reader. And in fact, you know, so it's very likely that it may still be disabled from the last time we told people to disable it. If not, the fact is, JavaScript in a PDF reader really seems crazy to me. I mean, I recognize that, yes, you could script your documents and make them do fancy things. But I don't know

anybody who does. So here you've got this dangerous feature turned on which allows the exploitation of this, that you almost certainly don't need. So it's very easy to disable JavaScript. Mine's turned off on all of my installs. That's just something that I do as part of setting up a new system. And I would imagine our listeners who have done this in the past still have it disabled, so they're not in trouble.

Leo: So it's enabled by default. But once you disable it, it will continue to stay disabled in upgrades.

Steve: Correct.

Leo: Okay.

Steve: And I'm trying to think now, when I was doing the research for this, there was a comic strip on a major newspaper site was being served from a third party. And I can't remember, it was like King Comics or King something. Anyway, the point is that users started complaining that their computers were getting infected from looking at these comics. And it turns out it was this vulnerability, this Acrobat Reader vulnerability was being exploited. The SQL injection exploit was used on this third-party comic provisioning server to install this malicious Acrobat content, this malicious PDF content, which was then being served to the unwitting newspaper as content from a third party and was exploiting this to take over people's machines and install malicious code. So...

Leo: As is often the case, you've got these blended threats where they take advantage of one flaw, this SQL injection exploit, to take advantage of another flaw.

Steve: Right. So on the Firefox and SeaMonkey front, everyone who's been keeping themselves current will probably have noticed that the 3.5 progression of Firefox recently went to 3.5.6.

Leo: Noticed that, yes.

Steve: And anybody who was still at 3.0 went to 3.0.16. And SeaMonkey has also moved to 2.0.1. There were seven sets of problems fixed, a bunch of things. Multiple errors in JavaScript and the browser engine which can cause memory corruption and potentially remote code exploits. A bunch of stuff with the media library. The liboggplay and the Theora libraries for audio and video both have memory corruption and integer overflow problems. They found out that there was a way that the location bar, the URL location bar could be spoofed. And in fact the vulnerabilities were being reported, which would allow a hacker to place an invalid URL up in the location bar so it would look like a legitimate site, to obviously allow for phishing and spoofing attacks of different kinds. There was a privilege escalation vulnerability, a problem with NTLM reflection, and then some scripting problems - Gecko ActiveX object exception messages and things. All of that's been fixed. But a bunch of stuff. So and because, of course, this is open source, the full exploit details are available through analysis of the source. So anyone learning about these problems can say, oh, you know, I'm going to go figure out how to exploit these because they've got access to the source code, just as the Mozilla team do. And so

this is one of those situations where, you know, you want to get yourself made current because we keep seeing instances where people are being - and we'll be talking about one next here - where people are still being infected by things that have been fixed. And Mozilla says [chuckling], "If users are unable to install the update immediately...

Leo: Yeah, yeah, here it goes.

Steve: ...they should disable JavaScript in Firefox until they are able to install the newest version of the browser."

Leo: Now, you're just looking for those now.

Steve: I just - I Google that expression.

Leo: Google "disable JavaScript."

Steve: That's how I supply my security news every week. But importantly, I did want to mention that here we are on the eve of a new year, at which Mozilla has said they are going to stop moving the version 3.0 Firefox forward. So this is time, probably. Now, if something really horrible surfaces that was in 3.0.17, which I guess is what we're about to get from them, one wonders if they wouldn't fix that still. But for our listeners it's probably time to move from 3.0 to 3.5 and continue following that into 2010 because they're saying they're going to suspend support for that. And you can really understand why they would have to.

Clearly, there's a huge common code base. The fact that they're always having to rev both together means that all the same problems are in both sets of code. But it's just a huge amount of extra effort for them to continue creating 3.0 versions. And 3.5 has been around long enough, as you pointed out to me when I was reluctant to move. Elaine, our illustrious transcriber, however, reported that she went to 3.56 and something bad happened, like she lost all browser functionality and was having to fall back onto IE. I haven't heard any other problems with it. And I'm current across the board.

Leo: As am I, yeah.

Steve: So whatever it is, it doesn't seem to be a widespread problem. But I haven't heard from her since last week when she said ouch, she got bit by upgrading. [All fixed! Elaine]

Leo: I get these questions on the radio show all the time. And one thing I say is, anytime you install anything there's always the risk that the installation can go bad, and something gets screwed up, and you're going to have a problem. It doesn't mean there's necessarily something wrong with the program, but some percentage of people that happens.

Steve: Well, and Leo, no two PCs in the galaxy are the same.

Leo: Right, right.

Steve: It's just amazing this stuff works at all.

Leo: I know.

Steve: I mean, it just...

Leo: I say that every show.

Steve: I've got all these computers. There's no two of them that are the same. They've got different chipsets. They've got different hardware. They're different ages. They've got different histories. And they're somehow managing to mostly stay alive.

Leo: We had a call on Sunday or something from a guy who went to Vista SP2 and just black-screened. His computer just rebooted and won't start. And I said, well, you know, there are millions of people who've installed this SP2 without it. So it's something about the way your system was that was unique. And there's no knowing what it is. You can't predict it.

Steve: No, no, no. And I was thinking about we were talking a week or two ago about the days of DOS, where you would install DOS, which meant three files. And you'd add a memory manager, oooh. And it had a config file. Okay, I know what all that is so far. Now it's just, you know.

Leo: But at least then when something went wrong you kind of, you know, you kind of knew what was going on. Now there's so many things going on in the background, so many processes, so many applications installed. Just look, you know, launch your Task Manager. There's...

Steve: No, don't do it. No.

Leo: There's dozens and dozens of things going on.

Steve: And I will say, being really particular about what's running can bite you sometimes. Because one of the things that I'll often do is I will launch Task Manager and look at the thing just scroll. And I'll think, okay, I've got to do something about this. So I'll go in and go through a big bunch of weeding out process. Or I'll go over to the Windows services panel and selectively turn off things that I just don't need. And one of my favorites is to turn off DHCP because all of my IPs and my network are statically assigned. And then what'll happen is I'll take a machine somewhere else and forget that I

disabled the DHCP service because I didn't need it, and I didn't want it running. And it's like, oh, why can't I get an IP address? Oh, I know why. But always, you know, there's a few cycles that are lost in remembering what it is that I did. So it's sort of a mixed blessing. We just sort of all limp along.

Leo: It's just the way it is. There's nothing we can do about it unless you want to go back to DOS.

Steve: Speaking of limping along?

Leo: Yes?

Steve: The Conficker worm reports, "We're alive and doing well, thanks for asking."

Leo: I'm glad it's - what? It sent us a postcard?

Steve: Many of our listeners have asked, hey, whatever happened with Conficker? Well, it just reared its ugly head, or heads, again. It took down recently an entire seven-hospital maternity and continuing care medical network in New Zealand. All 3,000 of the PCs within their network had to be turned off. And the hospital's lab, the main hospital's lab is currently running at about 10 percent capacity. And the hospital is only accepting patients in need of urgent care. I should say "hospitals" because apparently it's a seven-hospital network plus some other ancillary labs and outpatient clinics. And it ripped through their entire network and brought them down.

There are - reports vary, but there are somewhere between 5 and 15 million PCs infected worldwide. And the number ranges that large because finding them and counting them is not easy. But there's an interesting site that I've never talked about, ShadowServer.org, www.shadowserver.org, which tracks botnets and denial of service attacks and other things. Really interesting. If you look just down the left-hand column of the pages on the site, for example, there's like a botnet map that you can click on. And they're showing, in real time, the size and location of botnets. And they also have charts over days, weeks, months and years where you can see with some level of accurate reporting where these things are, where the botnets are, where the infected computers are. You're able to break it down. They've got charts and tables, you can break it down by country. It's believed that the largest infections of Conficker, which, as I said, is alive and doing well, are in Asia, all throughout Asia, where between 10 and 27 percent of individual ISPs' total routable Internet address space is infected.

Leo: Wow.

Steve: 10 to 27 percent. Now, what's interesting is, and this comes back to what we were talking about, about late patching, is that Conficker spreads using an RPC, a remote procedure call vulnerability in the Windows Server service that was fixed exactly 14 months ago, on October 23rd of 2008. Here we are about on December 23rd of 2009, 14 months later, and Conficker is spreading using something that Microsoft fixed 14 months ago. And it is now called by the security community without question the most prolific

worm that has ever existed.

Now, in looking back, I got a big kick out of the wording that Microsoft uses to describe the vulnerability. Microsoft's text reads, "This security update resolves a privately reported vulnerability in the server service. The vulnerability could allow remote code execution if an affected system received a specially crafted RPC request. On Microsoft Windows 2000, Windows XP, and Windows Server 2003 systems, an attacker could exploit this vulnerability without authentication to run arbitrary code. It is possible that this vulnerability could be used in the crafting of a wormable exploit."

Leo: Possible, huh?

Steve: Gee, that could happen.

Leo: I'm just looking at the ShadowServer.org stats for Conficker population. Going straight up, going straight up it's actually hit 7 million unique IPs in November. I mean, yeah, it could possibly. Might, just might.

Steve: You might have a worm would come out of this.

Leo: Could be.

Steve: It could be bad. We would recommend that you patch this.

Leo: Could be.

Steve: And then they say "Firewall best practices and standard default firewall configurations can help protect network resources from attacks that originate outside the enterprise perimeter." Well, okay. That's standard boilerplate that all of their messages have on it. And in fact that doesn't work in the case of Conficker because what we see happening over and over and over, I mean, this is, remember, a whole bunch of UK hospitals were taken down. I don't know what it is about Conficker and hospitals. But...

Leo: Yeah, we gave them a hard time because we said why, why, why would a hospital have its important computing systems on the public Internet?

Steve: Well, yes. And in fact remember that in one case it was the equipment running the operating theater.

Leo: Yeah, was online.

Steve: Was having these problems.

Leo: In this case, the New Zealand case, it was when they were updating. So they probably weren't online until they said, okay, now, everybody go online, get system updates. Which is just boneheaded.

Steve: Well, it's hard to say what's going on. In a report from the government site which talks about this, somewhat tail between their legs, they say that their best guess is that an employee brought an infected machine into the network. And that was my point about firewall configurations not helping because it's - Conficker spreads within a local area network where you don't have firewall defenses. So, yes, you can have a strong perimeter that's protecting you from the public. But anything that pierces that, whether it's a USB thumb drive or a laptop that's brought inside, anything, any way for it to get in, and then it just has a field day. And as we see it just rapidly infects all the machines.

Now, here's what's difficult to understand, is how could - first of all, we don't know how long the infection's been in place. So it might be that Conficker has been in this network but just wasn't noticed until recently. And in fact in terms of remediation of Conficker it's actually causing a problem that Conficker is not more damaging because the fact that it just exists to exist, and we've talked about that, too, you know, it's typical of viruses and worms, they're still sort of at the proof-of-concept stage for whatever reason. The fact that it's not really doing bad things means that there's sort of - it's difficult to commit heavy-duty resources and time to cleaning it out of networks where it's just not that big a problem for a lot of ISPs. It's like, oh, yeah, well, so our users are infected, you know, good luck to them. They're still paying their bills. Yeah.

Leo: Amazing.

Steve: Anyway, ShadowServer.org is a neat place to poke around for anybody who is curious about all that. And I got a note from Troy Starkey, who said - his subject was "I wish to thank you for your software." He said, "Hi, Mr. GRC. Firstly, let me commend you on your fantastic software." Oh, yeah. Okay, I thought I was reading the wrong one, the way it started. "Let me commend you on your fantastic software. It has repaired a few of my drives in the past, so I truly wanted to show my appreciation by finally purchasing your great software." No, I think I am reading the wrong one.

Leo: Oh. You get so many, it's hard to choose.

Steve: Wait a second. I can - yes, yes, yes. Okay, sorry. Peter Lilley says, "Hi, Steve and the GRC gang." And the subject was *YAWN* - YAWN in all capitals with asterisks bracketing it - "A SpinRite miracle so typical you'll think it's boring." And so he says, "Hi, Steve and GRC gang. I've been listening to the podcast for four-plus years, so I must have heard around 200 stories of SpinRite's miracles. Today my sister called me in a panic as her old laptop would not boot, bluescreening on every boot, with NTFS.sys errors. Of course there was no recent backup because there was, quote, 'nothing important,' unquote, on the laptop. Oh, yeah. Except the hundreds of photos documenting the lives of her twin five-year-old daughters. While she sounded more than a little panicked at the idea of losing all her photos, I have to say I felt a serene calm as I reached for my SpinRite CD. Feeling like I'd lived through this story myself...

Leo: Very confident.

Steve: "...before just a few times. So I popped in the disk and ran Level 2. Boring. SpinRite found problems, of course, and recovered the data. Yawn.

Leo: Naturally.

Steve: "The machine loaded Windows successfully on the next boot. Yeah, yeah, like was there any doubt? Steve, of course I'm joking, and I think this is marvelous. You must swell with pride every time you hear yet another story that data recovered miracles are so commonplace. Thanks for a fantastic product."

Leo: There you go. I don't think there's anything commonplace about it.

Steve: And thank you, Peter, for a terrific piece of email.

Leo: It's just another day in the office of SpinRite.

Steve: Yeah, well, that's what we do here.

Leo: We have 10 great questions from our great audience and that Steve...

Steve: We got good ones, yes.

Leo: Reminder, if you want to ask a question, best way to do that is to go to GRC.com/feedback and ask that question. And Steve will - Steve likes to research his answers and come up with the right answers. So the best way to do that is just to go to GRC.com/feedback. Are you ready, Mr. Gibson?

Steve: Let's do it.

Leo: All right. Question one from Tom Newman in Discovery Bay - which is up here in San Francisco Bay Area - how the bad guys keep one step ahead when it comes to malware. Kind of topical for our Conficker discussion. Hi, Steve and Leo. I heard Steve mention a number of times about what Google and other sites are doing to detect and flag websites that can harm your computer with malware. You mentioned a link from a listener that pointed to Google's Safe Browsing site that flags bad websites? This past month I've received three messages through Skype saying that my Windows - through Skype, well, there's a little - there's something a little suspicious right there, I don't think Skype is warning you about things - that my Windows software was infected, I needed to download a software patch to fix the

problem. The messages pop up on my screen with "Software Update" as the title. All three times I was running on my Mac. So he knew it was bogus. Each time I blocked the sender, but I still received more of these messages. I did a Google search of the linked websites, the one with the "patch" software, and Google didn't show anything wrong with the site. However, when I did a whois of the site, it showed me that they were created within one day of when I received the warning message. Of course, it's a cat-and-mouse game. The website in each message was different by one or two ending letters, and the Skype sender was also different each time. By the way, that's a good way to detect a malicious site is it's kind of a random URL; right? Because they're constantly changing these.

Steve: Yeah, like 318x.com that we talked about last week, it's like, okay.

Leo: Yeah. And then there'll be a 319 and a 319z and - the wording of the messages indicated they were all set by the message's indicated person. The whois records indicated these sites were owned by someone in the Czech Republic. So my point is you can't rely on Google or anyone else to keep you safe when it comes to malware. The bad guys are always going to be one step ahead by changing website URLs, email addresses, et cetera, so you really do need to be careful and practice safe browsing. He's written a blog article about this with a screenshot of the warning message on GeekNewsCentral. By the way, this site is owned by Todd Cochrane, who does the Podcast Awards. Thank you, Todd. Congratulations on your Podcast Award, Best Technology Podcast, Steve Gibson. And thanks for the great work you do. That's Tom Newman who does the FrogView, I'm sorry, FogView Podcast from Discovery Bay, California. His website is FogView.com. And I will put, and I'm sure you will, too, a link to the blog posting in our show notes. [http://www.geeknewscentral.com/2009/12/20/attention-malware-on-your-computer/]

Steve: Yeah. Anyway, I thought that Tom's point was very well taken, and a good one. It reminded me, though, of sort of what we understand already about antivirus software. That is, you know, it's, as you mentioned, it's a cat-and-mouse game. You can't be guaranteed that Google is going to flag bad things in the same way that you can't be guaranteed that your AV system is going to know about emerging threats that haven't been added to it yet, just as Google can't know about a website whose domain was registered the day before that there are probably no inbound links to so Google's bots aren't following along. And still in both cases that doesn't mean, in the case of AV or Google's protection approach, it doesn't mean that they're useless because many bad sites do persist for a long time, just as many viruses and malware do persist for a long time.

And, you know, there is Google. And, you know, we talked about Web of Trust reporting problems, and certainly the AV manufacturers are constantly fielding new instances of these bad things, updating their signatures as quickly as they can and really trying to stay ahead of it. Certainly we know that a lot of effective blocking is done. So all of this is very good. It's just that security is a moving target, and software is porous. And so while Tom is right, I would say all of this is good, and certainly better than not having it at all. But certainly he's also right in saying we need to practice safe browsing. And all the lessons that we've learned, like not clicking on links in unsolicited email, and also not believing that you need to fix your Windows software when you're running on a Mac is...

Leo: And when you get the message from Skype. No, in fact I say this on the radio show, again to quote the radio show, all the time, which is you can have antivirus software and antispyware software. It's a good thing to have it. But it's secondary line of defense. You are the first line of defense, and your behaviors.

Steve: Yup. And clearly, the fact that Skype is being used as the delivery mechanism is meant to catch out less sophisticated users.

Leo: Who probably don't even know they have Skype running.

Steve: Exactly. Or, I mean, again, many non-computer-savvy people are going to see a Windows Update box pop up and believe it. And because, I mean, because Windows also pops up valid boxes saying similar things. So, I mean, it's sort of a social engineering hack, you know, coming through a bizarre channel, i.e., Skype. But certainly some percentage of users are going to say, oh, wow, I guess I have a problem. Click here to fix. So it'll catch people. None of our listeners, but others.

Leo: Not just Skype. I mean, all IM clients are used this way. Every time I log into MSN I get 30 of these. And I get a lot of messages saying - from Pretty Girl 13 saying, oh, yeah, I'm all alone. And usually, by the way, those are not what you think they are. They're not a come-on to join some adult site. They're really a link to a page that is malicious.

Steve: Yes. Yes, they just want you to click...

Leo: They want you to go to that page.

Steve: Just click this one link, just this one time, is all we need.

Leo: So don't say, oh, I'm immune to this, let me just check this out, it's ha-ha. No, no, no, it has nothing to do with Pretty Girl 13. Here's a question that's a whole category of itself: "I read that this is broken. What's the story?" Rob near Ottawa, Canada wonders, "How long does it take to crack SSL?" Steve and Leo, I read an article on Mark Taw's blog [MarkTaw.com] - and I'll put a link again in the show notes, it's a long link. I don't know if this post is accurate or just FUD. The article claims that in 1995, SSL was brute-force cracked in 32 hours. Now with today's computers it can be cracked in a matter of minutes. Is that true? Was it true? Is it true today? Could you explain this? Help.

Steve: Well, it's an interesting blog posting because it talks about how SSL was cracked in a very short time 15 years ago.

Leo: Yes.

Steve: And we've talked about the SSL protocol. When I looked closely at the details of the posting, I saw immediately what was going on because they posted the key that was found that allowed them to decrypt the SSL connection. And it was exactly 10 hex characters. Well, a hex character represents four bits. So 10 of those is 40 bits. And those old-timers of us may remember when 40 bits was the maximum length of encryption that was exportable from the United States.

Leo: That's right. It was intentionally crippled.

Steve: Yes. Encryption was considered a munition, and so it was regulated by the State Department, and you couldn't export it from the U.S. So SSL could be forced to negotiate a symmetric key of only 40 bits.

Leo: Intentionally crippled.

Steve: Yes, exactly. And so remember that 32 bits is 4 billion. 32 bits is 4 billion combinations. So we're adding only 8 bits to the 32 to get 40. Well, we know that 8 bits in a byte, that's 256. So it's going to be 256 times 4 billion, which is a big number, but it's 40 bits. I mean, we know that's no longer strong enough.

Leo: Well, and it wasn't then. That was the whole point; right?

Steve: Correct. Even then you could put - if you put a bunch of machines on the problem. And so what I liked about this is it highlighted something, which is we've talked about the SSL protocol and all this fancy public key technology and certificates and certificate authorities and chains of...

Leo: Webs of trust and, yeah.

Steve: Certificate chains, exactly, all of this. Well, all of that ends up generating a symmetric key which is of a certain length. And that's actually used to encrypt the payload. So all that fancy protocol stuff is basically to authenticate the endpoints and allow them to negotiate a secret symmetric key. Well, 15 years ago that was - could be, in fact, you weren't limited necessarily, but if either endpoint could only support 40-bit SSL, then they would negotiate down to that. So back then the result was a symmetric key which was just too short. I mean, it arguably then should have been a minimum of 64. But it was deliberately crippled. So that meant that if anyone then captured traffic, then you could do an offline attack by just guessing, brute-forcing that 40-bit key until you decrypted the contents of one of the packets, and then you'd have the contents of all of the packets.

So it's one of the reasons why we really do need to future-proof our technology. I mean, for example, we've talked about this notion of what's safe today may not be safe tomorrow. Of course everyone talks about quantum computing and the idea that we're going to suddenly have a huge jump in computing power. It's not clear how that's going to happen or when. But if you were doing packet captures, like back then, and you were capturing packets that were 40 bits encrypted, you could certainly break 40-bit

encryption using today's technology just in a blink. I mean, there'd just be no problem doing it at all.

But I like the idea also, it's not something we've talked about before, which is that the upshot of all of the fancy handshaking is a symmetric key. And if that is not long enough, you can brute force attack it; and, you know, you get the contents of the packets. So to answer Rob's question is to be very sure everyone understands, 40-bit encryption is no longer in use, cannot be used. Thankfully, the laws changed in the U.S. that allow everyone to use much stronger encryption, and SSL now does. And again, even though it doesn't sound like 128 or 256 bits is that many more bits, oh, baby, I mean, when every single bit doubles, and you double and double and double, you know, what, 256 compared to 40, so you're doubling 216 times, that's really a lot of doubling.

Leo: So you couldn't - even, like, the NSA couldn't crack it at this point. In anything like a reasonable amount of time.

Steve: Everyone, I mean, you ask any crypto person who's at state of the art, they do not believe that 256-bit symmetric key encryption is crackable by anyone.

Leo: At all.

Steve: Everyone says, oh, well, yeah, but, you know, there's back doors and all these things. It's like, well, we've gone over the AES algorithm. We did a podcast on it, exactly how it works. People with absolutely no agenda, mathematicians and cryptographer who would love to put a feather in their cap by finding a weakness, have studied it like crazy. They've used reduced-strength versions to say, oh, well, instead of doing 14 rounds of encryption, if we only do five, then there's some things we can see where the bits aren't really being scrambled as deeply as they need to be. And it's like, yeah, but the designers knew that. That's why there's 14 rounds, because you very quickly lose all ability to track this. So everything we know says that nobody can decrypt this if it's well-enough encrypted.

Leo: Good news.

Steve: Yes.

Leo: Fear not. But it's funny because these articles, you know, bloggers can write anything they want.

Steve: Yeah.

Leo: And this guy's not a security expert.

Steve: And he asked, is this FUD?

Leo: Right.

Steve: The answer is yes. I mean, that's the history.

Leo: I wouldn't say it's intentional FUD, it's just it's misguided. The guy doesn't know what he's talking about.

Steve: Right. But I did like it from a historical perspective.

Leo: Absolutely, yeah.

Steve: Because it shows where we came from.

Leo: But we get these questions, I'm sure you get them all the time, from people who say, I read an article that said, you know, whatever, you know, public key cryptography's been cracked, or WPA2 has been cracked. And usually it's yeah, but that password or 40-bit key, that kind of thing.

Steve: Yeah.

Leo: You listen to this show, you'll get the straight stuff. But that's why we answer these questions. Scott in upstate New York points out a security hole he says you could drive a truck - or fly a Predator UAV - through. Steve and Leo, the people in the Pentagon should really listen to more Security Now!. This was in the news. It turns out that the video feeds from American UAVs - what is that? Unintended Air...

Steve: Unmanned Aerial Vehicle.

Leo: Unmanned Aerial Vehicle - are sent unencrypted to the ground. Insurgents discovered this and are using $26 off-the-shelf equipment to intercept the feeds and plan their operations around the locations of the drones. This flaw has been known since the '90s, but Pentagon officials assumed it wouldn't be exploited because, hey, those Afghans and Iraqis, they're stupid; right? This is a clear failure of the security-through-obscurity model. What do you think? Scott, a loyal Security Now! listener.

Steve: Well, a bunch of our listeners, as you might imagine, posted questions about this, wanting some commentary. And we're in a little bit of an information deficit, as one might imagine, because...

Leo: Yeah, they don't want to talk about this.

Steve: ...the government is embarrassed. I did hear an interview from someone who sounded like he understood the equipment. And as I understand it, the problem is that there is a large network of ground-based receivers which are more than a decade old.

Leo: Ah.

Steve: And the problem would be, while you could trivially, I mean, this is - I guess it's digital. It's digital video because they're using a digital video satellite software. There's some software made by a Russian company who is also, coincidentally, not happy that their software, which you can purchase for $25.99, that their software is being used. The idea is their software is - I think it's called SkyGrabber or something. The intention is that people who use satellite Internet links are vulnerable to having their satellite Internet monitored because the encryption is not very good. And this stuff apparently decrypts this digital information - if it's even encrypted, and it may not even be encrypted. It's just it's what's necessary to essentially demodulate the digital data and convert it back into digital.

So it turns out that the UAV planes are, while they do have encrypted command and control technology, so it is not possible for anyone to intercept and, like, take over the planes, for whatever reason they emit unencrypted video which is being received by these ground, very old, very inexpensive low-technology ground-based receivers, which collect the video and then relay it back to command central, wherever that is. And I've actually heard, I don't know this at all for sure, I know that the UAVs are built by a company in San Diego, Southern California. I've heard that they're being flown by pilots also in San Diego. So, I mean, they're - literally we're talking remote control. These things that are flying over Afghanistan and Pakistan are being controlled from halfway around the world.

Leo: I heard Las Vegas. But, you know, same thing.

Steve: Yeah, exactly. Way, way far away.

Leo: Yeah, yeah, here.

Steve: So here's the problem, is that they're inexpensive. It's old technology. But that's really also not an excuse because there's a new version I think called the Reaper which the government has just commissioned and ordered from the same company that also has unencrypted video. So it really does seem to be a case of people just not paying attention to this and not thinking it's going to be a problem.

What happened was that some Shiite terrorists were captured, and their laptops were found to contain recorded video from the UAV over-flights. And then that was - I think that was in '08. So more than a year ago that happened, but nothing was done to change it. And the problem is, the infrastructure that exists, the equipment which is installed can't be upgraded. Sure, you could encrypt the feed on a UAV which costs an amazing amount of money. And we know that putting even just a little symmetric cipher in there would be zero overhead. The problem is, apparently the receiver needs to decrypt it. I guess it's unable to forward it encrypted back to home base.

I mean, again, we have no idea of all the specific details. But it does seem to be the case that for whatever reason the existing infrastructure can't take encryption without a huge reinvestment, which is much more than is in the budget right now. And what's really dumb is that the various people who are embarrassed are saying, oh, well, we really don't think it represents a big problem. It's like, okay. Except that for people who know the terrain, the people on the ground who are adversaries, they're able to look at these video feeds and know exactly where the plane is flying over. So that seems to me to be a problem.

Leo: It's an interesting story, isn't it.

Steve: Yeah.

Leo: Question four, Chris in Texas worries a bit about the free SSL certificates you might have seen: Long-time listener, but can't seem to remember if you've covered this already. There's a company called StartCom Ltd. that gives away basic SSL certificates. Hey, that's cool. I'm just wondering if these certificates are safe to use. They're listed as a trusted authority in Firefox and IE and all the other browsers. But free, what's the catch? Is there one? I've gone through the process of signing up in which they generate you a certificate signing request for authentication, not a username, and validated my domain ownerships. Everything has seemed professional. No process has varied from authorities I've used. What's your opinion on these guys? Your show has been part of my commute since the days of single digits. Look forward to my feeds every week. Thank you for all the wonderful information. Well, there have been other free certificate authorities in the past.

Steve: Well, okay. So here's the deal. StartCom is a good little reputable certificate provider. They do have their certificate authority certificate installed in all the major browsers. So if you look at the root certificates that are installed in IE, StartCom is there, and the same thing for Mozilla. So it is the case that their certificates will work, meaning that if someone were to use a free certificate from them for their server, then all the users who are using popular browsers would be able to establish SSL connections without any problem.

The only downside is if it were for some reason really important to trust this company, that is, to trust your connection, a user that is on a client, on a browser, might check the certificate chain and think, huh, this certificate was signed by StartCom. Never heard of them. Heard of VeriSign, heard of GoDaddy, don't know who StartCom is. So remember that implicit in this is not only that we're using them for encryption, but we're using them for authentication of the server. And that's important in this era of DNS spoofing and man-in-the-middle attacks and all that kind of stuff.

So I would say to Chris there's nothing wrong with these at all. If their root certificates were not installed in all browsers, then you'd have a problem that some users of browsers that didn't have the StartCom root certificate wouldn't be able to make a connection because, you know, the signer of the certificate that's being offered by the web server wouldn't be known to the browser. But it is. So that's not a problem.

The only issue would be that, you know, if a user was concerned about the reputation of who signed the website's certificate stating that we verified all this. StartCom's FAQ pages do talk about this and explain that, you know, they have a highly automated

system which allows them to issue short-term, that is, one-year free certificates. My guess is that their model is they're issuing certificates for a year. That'll bring you back every year. And maybe you'll decide, hey, this has been working for a few years. I think I'm going to pay for a longer length certificate, where they will do a little more testing in order to justify the longer life of the cert. But I think it looks like it's a fine solution.

Leo: Great. Yeah, I mean, there have been a number of these companies. I mentioned Thawte, which was purchased unfortunately by VeriSign.

Steve: By VeriSign.

Leo: Stopped doing the freebies. But I've used free certs for email and stuff, for PGP, and I don't see anything wrong with that.

Steve: Nope.

Leo: It doesn't cost them that much to do a cert, let's face it.

Steve: Costs them nothing. It's only the bureaucratic side of verifying identities and domains and so forth.

Leo: It's been my experience that free certs sometimes don't go through as many hoops to verify.

Steve: Precisely.

Leo: Yeah. Anton Wirsch in Tokyo, Japan, has some great info about San Diego's Sea World fingerprinting at the entrance. This is not just Disney. In late November I was home for the holidays and went to Sea World in San Diego. After I had purchased my ticket I found at the entrance to the park they were electronically fingerprinting everyone as they entered. After listening to your podcast, this concerned me. Who was managing the fingerprinting data? What security measures did they have? What happened to all the data in case Sea World went under?

I was instructed to a window where I could ask questions about the fingerprinting. The young person at the window assured me the fingerprint is not stored anywhere. He showed me a paper that explained what their system was doing. The fingerprinting machine used the fingerprint to generate a number that was then mapped to the barcode of the entrance ticket. This allows Sea World to ensure that only one person is using an entrance ticket. If a person leaves the park, then wants to reenter, the fingerprinting machine should regenerate the same number, match the mapped barcode on the ticket. My concern, though, is that I just have to take their word on that, that that's what they're doing. I have no proof they're not sharing my fingerprint or storing it in their database. I'd like to hear your thoughts on this. That does make sense. That's a reasonable way to use it.

Steve: Actually it's very cool. I like that a lot. I have several thoughts. One is it's terrific that there is a window where you can go and ask your questions, and that they have a paper that they will show you that explains this. I mean, it does tell me that Anton who listens to Security Now! is not the first person to feel a little sketchy about why Sea World wants his fingerprints.

Leo: Right.

Steve: And their solution I think is very clever. Essentially they're hashing his fingerprint. They're taking details which are probably hopefully relatively static, and they're running it through a hash to essentially generate a digital signature. But as is true of all hashes, that's inherently a one-way function. That's what a hash is, when we've talked about it, and we've discussed hashing at length in the past, the idea being that you can take a document of any length or a blob of information, you run it through a hash, and it produces a unique value for that document.

Now, what's special about that is that you cannot get the document from the value. So in this case you cannot get someone's fingerprint from the number that results from hashing the fingerprint. Yet you can still use it for identification, that is, the same fingerprint should generate the same hash value. And what you would like to do is to limit the length of the hash. That is, if you took a fingerprint and hashed it to a really large multi-bit hash, then essentially you've got something as good as a fingerprint, that is to say, if no two people could ever be expected to have their fingerprints hash to the same value, then what you've got can still be used to uniquely identify you. So that's not what you want. What you want is a limited bit-length hash.

So, for example, imagine that the fingerprint were taken and hashed, and only 16 bits were kept. You could take any 16 bits out of the, you know, a 256-bit hash. So now what do you have? 16 bits, as we know, is 64K. So now you've turned a fingerprint into one of 64,000 possible values, meaning that many people on the planet will - because there's a lot more than 64,000 people. That is, everybody's fingerprint will produce a number, randomly distributed if it's a good hash function, between 0 and 65535. So it's no longer sufficient length to uniquely identify a person, yet it is sufficient to disqualify a person who's impersonating someone else. That is, the chance would be one in 65536, one in about 65,000, that an arbitrary imposter would have a fingerprint that hashed the same number.

Now, I don't know that Sea World is using a truncated length hash. If they were, that would be very cool. They'd get the seal of - they'd get the Security Now! seal of approval for their technology. But, you know it's nice that we're seeing evidence of, you know, this kind of responsibility. Because I think it's entirely acceptable that the fingerprint itself is not being stored, but it's being turned into something which is a unique token that represents the person. That's cool. And as for believing them, I absolutely would believe them. I would say they've got a window, they've got a paper, you know, it seems unlikely that they're going to produce all of that and explain it all and then have it not be true. If nothing else it would open them to a lawsuit.

Leo: Oh, gosh, yeah. You don't want to lie about that.

Steve: Exactly.

Leo: Yeah. Marv Schwartz, who's at the great Case Western Reserve University in Cleveland, one of the excellent tech schools in the country, reminds us about blocking lookups with the HOSTS file. He says: Steve, way back in Episode 42 you told us about blocking unwanted sites with a hosts file. From what you've said in 227 - our last episode - wouldn't blocking 318x.com with a hosts file entry now be a very good idea? If your hosts file blocked it, then 318x.com would never get control and therefore could not invoke chained malware. Thanks again to you and Leo for a wonderful program which I enjoy without fail every single week. And congratulations on winning the Best Technology Podcast Award. You deserve it. Best wishes for the holidays and New Year. Marv.

Steve: Many of our astute listeners mentioned that. So to all of them, and to Marv, I wanted to say absolutely right. Remember that we talked about this, and this is an ongoing threat, and that it is as a consequence of that SQL injection a huge number of sites have been infected with a little embedded iFrame which contains a reference to 318x.com. So the first thing your browser that encounters one of these pages will do is attempt to pull the contents from the URL contained in that little embedded frame, that little iFrame. To do that it has to look up the IP address of 318x.com. If you stick that in your hosts file, the search stops right there. Nothing will go any further, and you are safe. And lord knows nothing good is ever going to be at 318x.com.

Leo: No reason to have that.

Steve: It's not like you're going to not be able to send email to your mom or anything for Christmas.

Leo: But as you pointed out, it's very likely to be 3189.com or whatever at some point. So it's only a limited, of a limited value.

Steve: Yes. And in fact in the reporting, in the research that I did prior to last week there are variants of that already in existence. So again, blocking - but which is not to say that blocking 318x.com is not a - would be a bad idea. Putting it in the hosts file is a good idea. I wish I had suggested it last week. So I wanted to give everyone some props for saying, hey, that's a great place for the hosts file. Absolutely is.

Leo: Another Ohio resident, Matt, tells us of a new finger-based biometric technology, not fingerprints. Steve and Leo, I heard discussions on your podcast about banks wanting fingerprints and various retailers now requiring them. But where I work we utilize a slightly different system. We sell memberships for various things. We used to give out cards with pictures on them, but it costs too much money and time to have card printers and to print them. So we have a new system that reads finger veins. From what I've seen it works extremely well. They claim near zero percent failure rate. So now we don't have membership cards, they just scan their finger. People don't have to worry that their fingerprints are on file, just their veins. And what are people going to really do if they know the layout of your finger veins? And he points us to m2sys.com. They make the finger vein reader. Well, this is similar to the palm readers, right, that you use at Level 3, or your

provider uses.

Steve: Well, it's a different technology. The palm readers are measuring physical, sort of external physical metrics. They're, like, measuring the length of my fingers and the size of my hand and so forth. This is actually, instead of a reflective technology, which is what a fingerprint uses, this is a transmissive technology. That is, you put your finger into this thing. And it uses near infrared light. So the infrared light shines through your finger from the back of your finger down to the front, where there's an imaging array which sees it. And hemoglobin absorbs in the IR. So as a consequence of that, the imaging scanner can see the layout of your veins. And this is actually a technology that's been known and used for years in a different way. We've probably seen pictures of people in hospitals who have something slipped over their finger which is able to take their pulse and measure their blood oxygen by doing exactly the same sort of thing. So I think there are even some things you can clip on like an earlobe in order to, like, keep track of pulse because as the veins throb with our heartbeat, that changes the amount of IR which is absorbed in the path through that chunk of flesh, whatever flesh it happens to be. And it's very easy to get a pulse from that. These guys are going a little further and actually getting an image from it.

So on their site they talk about the advantages of this. For example, one of the problems with fingerprints is they, being on the surface, they are prone to damage. You could get a - you could have a scar or cut yourself or burn yourself, you know, various things that affect the surface of your finger do not affect and do not change over time the layout of your veins in your finger, which they claim is every bit as unique as the print on the surface of your finger. So I just thought it was an interesting little tidbit. People may, if this catches on, see a different kind of technology in the future where they're sticking a finger into something rather than on something, the idea being that IR light will be shined from the back of your finger through your finger to pick up a vein pattern.

Leo: Interesting.

Steve: Well, and it has the other feature that it isn't something that you leave on water glasses in restaurants and things.

Leo: That's a good point, yeah.

Steve: Yeah.

Leo: Yeah. So it doesn't have any value in criminal investigation or that kind of thing.

Steve: Right. From a forensics standpoint it's not going to help you. But it certainly would, it's the kind of thing you probably know is going on because you're having to stick your finger in a hole in order to get it read.

Leo: Question eight, Brian Kuner in Akron, Ohio - three Ohios in a row, Steve, did you do that on purpose? - just discovered that GRC's unreleased DNS Benchmark fixed a longstanding problem. You haven't released the DNS Benchmark?

Steve: Nope, still working on the documentation.

Leo: Oh, okay. Because we have been able to download it and use it.

Steve: Oh, absolutely. And it's done. It's completely finished. I just need to get all of the screens documented.

Leo: Publicize it. Just wanted to let you know that I'm a Security Now! listener. Just listened to 226, heard you and Leo talk about the DNS Benchmark utility. So I downloaded it and ran a test. My router DNS tested very low on performance, even though I'm using OpenDNS. I've been having a problem for a while where I open Firefox, and it has problems reopening about half the tabs I'd opened previously. After looking at the benchmark results I changed my computer to directly point to OpenDNS, and quick testing has shown it solved my Firefox problem. I could kick myself for not trying that before, since I should have known better. Thank you for writing this utility and resolving a longstanding problem I've been having. Why would a router be slow? I guess just kind of a dumb router, I guess, huh?

Steve: That's exactly the problem. And in fact I'm at a loss, complete loss to understand why routers have decided to take on the responsibility of proxying DNS on behalf of their LANs. It's not like they're adding any value at all.

Leo: Right.

Steve: They're not caching. These things barely have four bits to rub together. I mean, they are, you know...

Leo: They're dumb. They're dumb little boxes.

Steve: They're an empty plastic box with nothing inside. And it's like, oh, well, we're going to...

Leo: I'd like to help.

Steve: Yeah, we're going to offer DNS services to all the machines on the network. It's like, why?

Leo: Do most routers do that, or no?

Steve: It seems to be on the rise. It's a new thing. Older routers didn't. They would just pass through whatever DNS servers the ISP upstream was giving them. Now, for whatever reason, they advertise themselves, their own gateway IP, as the sole DNS for the network. Which then requires that they field these DNS queries. And the problem with that is that there's some sophisticated logic that is employed by Internet-aware computers. From the very dawn of time UNIX machines were on the Internet where, you know, the requirement was two DNS servers.

And of course then Windows and Macs and Linux machines, all Internet-aware systems, generally have pretty good logic. They will recognize which DNS server has been working well and issue requests to that one. Then, if it seems to not respond, they'll try again. And if it still doesn't respond, then they'll send requests out to all the other DNS servers they know about. And you can actually register more than just two DNS servers. You can, if you use like the advanced configuration tab, at least in Windows, and I'm sure it's the case in UNIX and Linux and probably is on the Mac, you're able to equip your machine with more backup DNS servers. In which case it'll send parallel queries to everybody else it knows about. And the one that responds fastest then becomes - moves to the head of the list, and it'll be the one the system uses.

So it's very intelligent. I don't think any of that is going on in these little routers that can't get out of their own way. Who knows what they're doing? I mean, they're apparently doing the least possible. And what we discovered, one of the reasons I've ended up being so down on routers imposing themselves, is that I have an also as-yet-unreleased, but also finished, but not yet documented, spoofability test, which is very cool. And the DNS Benchmark is part of the release of a whole suite of DNS things that I'm getting ready to do. It crashes these routers. That is, they get in the way.

And I'm just doing regular DNS things, and we found about 10 or 11 so far that are completely crashed by this. Just by doing DNS lookups. It's like, get out of the way. They're offering no value. I do not understand why it's become so popular. No one that I've spoken to has even been able to offer a reason. If it were caching, for example, if the router had some weight behind it, and it was using a good upstream algorithm, and for example providing caching services, then you could argue that other machines on the network could, when they want to go to Google, the router would know, instead of them having to cache it themselves. But these things don't. They're just pitiful little empty boxes of plastic.

So anyway, it seems like - I'm not surprised that Brian found that getting his router out of the way helped him. And I thought it was worth sharing with our other listeners. Probably by the time we talk about how we're able to crash routers, nobody who listens to this podcast will still be using their router as a DNS server. At least I can hope that because it seems like a really dumb thing to do. But it is the default configuration in many cases.

Leo: Interesting. All right. Long one. Take a deep breath. This is from Ilari Kajaste - I'm sure I'm mangling his name because it's in Finnish.

Steve: You do a much better job than I would, Leo.

Leo: Finnish, often the letters in the name have no relationship to the pronunciation, it has been my experience.

Steve: And fortunately Elaine also receives this PDF. So she always spells everyone's name correctly, even if we mangle the pronunciation.

Leo: She's good at that. I'm going to say Ilari Kajaste - because I know "j" is "ye," is almost a "y," Kajaste - in Finland takes issue with fingerprints and other biometric data as identification. He says, or she says: You've been talking a lot about fingerprint privacy on Security Now! lately. I'm a very strong privacy advocate myself, member of the local EFF - as am I - and all that. But I can't help thinking there's something very odd in the talk about biometric data privacy. The problem is this: A water glass I handle at a restaurant, the door handle I touch when entering a store, the kiosk I use for ticketing for a bus, aren't these all valid places to surreptitiously pick up my fingerprint? No matter how hard I protect my precious index finger's unique pattern, I'm essentially giving it out every day. Same extends to all biometric data. Well, not all. I mean, I don't know how people would get your iris.

But with technological advances, DNA will become very inexpensive to analyze; right? Iris patterns can be obtained from a distant photo - oh, I didn't know that - and so on. By our simple existence we are essentially continually broadcasting biometrics to the world. Considering biometrics is something that's supposed to be private, seems like looking at the whole issue upside down.

I think we really need to start considering biometrics - fingerprints, iris patterns, DNA, all of it - not as private, but rather as public data, safe to be broadcast even on one's own home page. That way we, society, could learn to use them reasonably as we'd stop considering the data itself private. Biometrics would still remain as valid use for identification for many purposes. There is still the hassle with reproducing biometrics even if you have the data. Duplicating a fingerprint from an image isn't all that easy. Duplicating iris patters or DNA is much more difficult. Especially when using multiple biometrics combined and verifying the read event by measuring say, body temperature and so on, fooling the biometric readers can become quite a problem. It's a matter of balance, of course. Biometrics should only be used where the cost of counterfeiting is higher than the benefit. But the same goes for all security measures. Sure, it becomes easier to fool the readers with technological advancements. But the biometric readers will improve as well to counter that. However, keeping biometric data private will only become increasingly difficult. It's an uphill battle that should be abandoned. I'd be very, very interested in hearing your thoughts on this, Steve, as I really appreciate your thought-provoking insights into security matters. I've been listening for quite a while now. I'm currently working my way through most of the Security Now! archives, truly an amazing resource. Thank you, Ilari. That's a good point, isn't it, Steve.

Steve: Yeah, he really, I mean, it is absolutely a good point. The other thing I was thinking about as I was listening to you talk was voiceprint analysis.

Leo: Sure.

Steve: There's another example. I mean, you and I are literally broadcasting our voice all over the place. So if we were ever depending upon the non-reproducibility of our voice, that would be a problem since there's an amazing amount of sample of our voice now. And I like his point about the fact that it's not the knowledge of the biometric, it's its ability to produce it on demand.

You know, for example, if law enforcement needed a DNA sample, you know, they don't accept one from an envelope from you. They say, "Say ah," and they swab the inside of your mouth, I presume, or at least that's what they do on the various shows that I've seen. So they're getting a live real sample from the person that they've identified the sample as coming from. So you have no ability to spoof the DNA that they're acquiring at that stage.

So everything that he said aside, am I going to post my 10 fingerprints on my website? Uh, no. It's, yes, it's the case that I'm leaving my fingerprints around. But all of the security that we use is shades of gray. You want to control access to your DNA and to your fingerprints, and now we know to the vein prints inside our fingers, to whatever extent possible. Even though Sea World is hashing a fingerprint, I'm glad that our listeners, I mean, I don't think it's wrong for our listeners to be a little skeptical. And I love it that they're asking questions, as clearly other people are, saying wait a minute, why do you want my fingerprints? I'm concerned about privacy. I don't want to be giving them out all over the place. I agree with him that, you know, spoofing them is a problem. But at the same time I don't think there's any substantial cost to minimizing disclosure. I think that's still a useful strategy.

Leo: Yeah. Yeah. Last question from Marco Silva in the Madeira Islands, Portugal.

Steve: It's nowhere near Ohio.

Leo: No, nowhere near Ohio for the last two. I love how, you know, just totally global our audience is. He wonders about your unreleased, another unreleased program, the Router Crash Test, and wonders if router configurations could be changed. Steve and Leo, I just ran the "Router Crash Test" on your DNS website. Fortunately, my router didn't crash. But if I run this test on a router that does crash, could doing so change the configuration? You say that the crash "never results in any permanent damage to the router or other equipment. Some routers reboot themselves and restore their service. Others hang and need to be powered off for a moment and powered back on." He's just worried whether this test could accidentally change the router's configuration on a router that has crashed. I don't even know about this, by the way.

Steve: [Laughing] I'm behind. I'm working as hard as I can to catch up.

Leo: What the heck is this?

Steve: We will be doing podcasts about this stuff as soon as I get all of the pieces assembled. What we discovered was that it was possible to crash routers. And we then decided, "we" meaning all the denizens who hang out in the GRC newsgroups and I, when we were working on this about a year ago, actually. And this all resulted from the

Kaminsky DNS spoofing stuff that was in early '08 or mid '08, I guess. So we then determined, well, we had a bunch of users whose routers were crashing when we were developing the spoofability test. So we figured out how to keep that from happening, but then also what it was that they weren't liking. And this is exactly what we were talking about a second ago, about the routers that are imposing themselves as DNS servers that were unable to field valid DNS queries and replies.

And so I created a separate test. I made the normal spoofability test not typically crash routers, and then created a router crash test because as all of our listeners know, what starts out as a crash today is a remote takeover exploit tomorrow. And the concern is that it might be possible to send DNS replies at routers that are proxying and take them over remotely. So, I mean, we have no evidence that that's the case, but that's the way all these things start.

Leo: Sure. Anytime you've got a crash, that's the potential.

Steve: That's the potential. And essentially we are able now to remotely crash people's routers that are imposing themselves in the DNS lookup process. So to answer Marco's question, routers store their configurations in their firmware, in nonvolatile memory. We can't, since we're crashing the router, we don't know for sure that we're not altering the configuration. But it seems really a stretch that we would be, and we've never seen a situation where we have.

Leo: But use at your own risk. Don't just go around crashing your router for fun.

Steve: [Laughing] Correct. And I do list the 10 or 11 routers that are known to be crashable at this point. And every so often somebody reports another one. It's like, okay, here we go. I mean, they're just, you know, these things are inexpensive consumer appliances, and they're being pushed beyond, I think, what they should be doing by interposing themselves in DNS. They're just - they're not adding any value for doing so.

Leo: Steve, always a pleasure. You make everything seem so clear. And today I wasn't too frightened.

Steve: Good.

Leo: Which is nice.

Steve: Good. Because it's the holidays, Leo. We don't want to frighten you and Santa Claus.

Leo: Merry Christmas. Have a great Christmas Eve and a safe trip to see your mom. And we'll talk again next week. For those who watch live, we're doing two shows next week because the following week I'm going to be out of town at CES, covering the big Consumer Electronics Show. And Steve is committed never to missing an

episode. So two shows in our live taping next week. We'll start early and go late.

Steve: Perfect.

Leo: Thanks, Steve. Have a happy holiday.

Steve: You, too, Leo. Happy Christmas, Merry Christmas, Happy Holidays, all that, and talk to you next week.

Leo: We'll see you next time on Security Now!.