
# Listener Feedback #145

**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-356.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-356-lq.mp3

---

SHOW TEASE: Time for Security Now!. Steve Gibson is here. We've got a lot of security updates, including news about LinkedIn, Flamer, Stuxnet, and your questions. It's all coming up next on Security Now!.

**Leo Laporte:** This is Security Now! with Steve Gibson, Episode 356, recorded June 6th, 2012: Your questions, Steve's answers, #145.

It's time for Security Now!. Couldn't be a better day to cover security with Mr. Steve Gibson, our Explainer in Chief. He's here today from GRC.com, and it's a Q&A episode, so we've got a dozen great questions from our audience. Good day, Steve.

**Steve Gibson:** Hey, Leo. Great to be back with you again, as always.

**Leo:** Oh, yes.

**Steve:** We've had an eventful week, not surprisingly. I think it was Friday that I tweeted to my Twitter followers the news that an investigative journalist with The New York Times had uncovered a multiply sourced report that one of his first acts after being inaugurated in office was that our Barack Obama, President of the U.S., ordered a speedup in the waves of cyberattacks that the U.S. was waging against Iran. So we have some news there. Of course that comes on the heel of last week's first opening discussions of Flame, what's now being called a "super cyber weapon" by Kaspersky, who's been looking into it further. We've got a bunch of information about that, new and interesting things. I was tweeting a lot of links as I was setting up the notes for the podcast. So anybody who wants links to these things just check my Twitter feed, @SGgrc, and you can get links to

these various important things. But we have a bit of sadness...

LEO Yeah.

**Steve:** ...in the sci-fi arena. Last night we got the word from his daughter that Ray Bradbury passed away at the age of 91.

LEO 91. I mean, that's a ripe old age, you know?

**Steve:** Yeah. He looked good for 91. He, of course, was I think probably most famous for "Fahrenheit 451," which was just an amazing book at the time.

**Leo:** "Martian Chronicles," too, I think, yeah.

**Steve:** Yes, and also he wrote "Something Wicked This Way Comes."

**Leo:** Oh, what a great book that is. Short stories. Lot of short stories.

**Steve:** And he's credited with being more of a literary influence on the genre. He disliked the term "sci-fi." He considers he only really wrote sci-fi, he considered, was his "Fahrenheit 451" story. The rest he considered more, you know, flights of fancy.

**Leo:** Yeah.

**Steve:** But his writing was so good that he was trying to bring, again, more of a literary feel to science fiction.

**Leo:** I think he's one of the great science fiction authors, and this is a great moment, an opportunity to go back and reread. I know that a lot of geeks I'm following on Facebook and Twitter said, oh, I'm going to reread "The Martian Chronicles" or "Fahrenheit 451." One of my favorites, and I refer to it a lot, is "The Veldt." That's a great one. In fact, they have a - I didn't see this, so I might have to download it - a dramatization of "The Veldt" on Audible.com. "The Illustrated Man," what a - and you're right.

**Steve:** Oh, yeah.

**Leo:** These are not really sci-fi.

**Steve:** Right.

**Leo:** They're fantastical tales.

**Steve:** Right.

**Leo:** "The Martian Chronicles" is; "Fahrenheit 451" is. Both of those, just fantastic. Boy. He's a great storyteller, one of the best writers. And I think one of the reasons he defies genre is because he's such a good writer and really goes well beyond what we're normally used to in science fiction. A truly...

**Steve:** And sadly, now we need to use the past tense when referring to Ray. But certainly not his work. His work will live on forever.

**Leo:** And a great supporter of science. And I think Ray Bradbury, an inspiration to many scientists for their work. I think that often we've found out that scientists look back to what they read, like "The Martian Chronicles" or "Dandelion Wine" or whatever and...

**Steve:** Were really influenced.

**Leo:** ...said that inspired my work.

**Steve:** Yes, yeah. This Friday is going to be June 8th, two days from now.

**Leo:** Yes.

**Steve:** It's the release in the theaters of "Prometheus."

**Leo:** Oh, I'll be in line.

**Steve:** Oh, goodness, yeah.

**Leo:** I'll be in line. This is the prequel - you know, it's funny, I know a number of people who didn't realize this - the prequel to "Alien."

**Steve:** Yes. And brought to us also by Ridley Scott, as was "Alien." HBO has a short, 15-minute presentation called "Prometheus: First Look." I've not seen it, but my TiVo sucked it in this morning at 9:00, or noon on the East Coast, and I know that it's, through HBO's schedule, it's scattered around. So anybody who's interested may be able to find it coming up. And I only - a buddy of mine told me about it. And apparently, one of the takeaways he had was that the set of the new James Bond movie, which is in production, was like a postage stamp compared to the set of "Prometheus."

**Leo:** You could tell it's a giant soundstage, yeah.

**Steve:** It is a huge, actually I think it is the largest set of soundstage environment ever made. Or, I mean, there's something like that about it. I didn't, I haven't, again, I haven't seen this.

**Leo:** Just the trailer, when you see the trailer, you can tell they're in a very large space.

**Steve:** Yeah.

**Leo:** Yeah. Cannot wait. Cannot wait.

**Steve:** Anyway, so I'm very excited.

**Leo:** Yeah, this is going to be an exceptional movie.

**Steve:** And again, just anecdotally, the phenomenal scene that, you know, that branded everyone who saw it in Ridley Scott's first movie, "Alien," was the classic scene of this creature erupting from the person's chest. You know, we'd never seen anything like that, ever. And apparently this movie has something different, but even more so. So...

**Leo:** [Laughing] I don't know if I'm ready for it. That was so terrifying.

**Steve:** It was just amazing.

**Leo:** Now, of course, Sigourney Weaver, who was the star of the first two, will not be in this one, I'm sure. Who is the - who are the stars of this?

**Steve:** Well, we have Charlize.

**Leo:** Charlize Theron. Oh, she's wonderful.

**Steve:** Yes, yeah.

**Leo:** One of our best actors. Boy, she's good.

**Steve:** And we have another android. And in fact, over on the YouTube, collection of YouTube videos for this, they're linked from the IMDB article, there is a full-length commercial that Wayland, who is the corporate interest behind all of this - remember, they were the people that did terraforming on "Aliens." Anyway, they have an ad for their android, where he's demonstrating himself and so forth. So...

**Leo:** Oh, wow. They're smart. I'll tell you, they've gotten very smart about marketing this stuff, haven't they.

**Steve:** Yup, yup.

**Leo:** They just - they've got a series of - this isn't - there isn't just one. It turns out there are many ads from "Prometheus." Wow. This is interesting. Look at ads for all of their products.

**Steve:** Wow. I haven't...

**Leo:** They're so smart about - they're so smart about doing this stuff now. They put a lot of energy into these...

**Steve:** Well, apparently Ridley Scott decided he was going to try to outdo himself. And from everything I've heard, this new one is going to be a contender. So I think it may be the movie of the summer.

**Leo:** Yup. Well, there you go. Good and bad news in sci-fi.

**Steve:** So, yes. New York Times' David Sanger put together an article. The title was "Obama Order Sped Up Wave of Cyberattacks Against Iran." And I'm just going to read the first few paragraphs, which since this is well written - it's a long story. It's five pages on their website. But this was properly written, so all of the meat is at the front. So he wrote:

"From his first months in office, President Obama secretly ordered increasingly sophisticated attacks on the computer systems that run Iran's main nuclear enrichment facilities, significantly expanding America's first sustained use of cyberweapons, according to participants in the program. Mr. Obama decided to accelerate the attacks - begun in the Bush administration and code-named Olympic Games - even after an element of the program accidentally became public in the summer of 2010 because of a programming error that allowed it to escape Iran's Natanz plant and sent it around the world on the Internet." Which is interesting. This is stuff we did not know before. Prior to that it was contained there, and it got loose.

So, "Computer security experts who began studying the worm, which had been developed by the United States and Israel, gave it a name: Stuxnet. At a tense meeting in the White House Situation Room within days of the worm's 'escape,' Mr. Obama, Vice President Joseph R. Biden, Jr., and the director of the Central Intelligence Agency at the time, Leon E. Panetta, considered whether America's most ambitious attempt to slow the progress of Iran's nuclear efforts had been fatally compromised.

"'Should we shut this thing down?' Mr. Obama asked, according to members of the president's national security team who were in the room [with him]. Told it was unclear how much the Iranians knew about the code, and offered evidence that it was still causing havoc, Mr. Obama decided that the cyberattacks should proceed. In the following

weeks, the Natanz plant was hit by a newer version of the computer worm, and then another one after that. The last of that series of attacks, a few weeks after Stuxnet was detected around the world, temporarily took out nearly 1,000 of the 5,000 centrifuges Iran had spinning at the time to purify uranium."

And so that's the start of a five-page story. Again, you can easily find it at The New York Times, or I tweeted it Friday of last week.

**Leo:** Where do you stand on this? We had a good debate on TWiT on Sunday.

**Steve:** Well, and actually one of our questions that we are going to come to was a listener who poses that. So I thought we'd maybe hold off...

**Leo:** Good.

**Steve:** ...discussing until we get to the question because I do, exactly with you, Leo, I wanted to discuss, I mean, the controversial nature of it.

**Leo:** Right.

**Steve:** Okay. So, Flame update. I also tweeted early this week, shortly after Microsoft released it, there is a very important update which Microsoft produced in an emergency, out-of-cycle release. It's small. It's 91K. At least it was in the case of my Win7 box, which I updated the moment I turned it on here to run Skype. Because it turns out that the components of Flame were digitally signed by Microsoft certificates.

**Leo:** [Laughing] Faux certificates? Phony certificates? Real certificates?

**Steve:** No, real certificates.

**Leo:** That's not a good way to hide your tracks.

**Steve:** Now, I mean, and when I first heard this, I was thinking, you know, I wonder if this wasn't arranged. I mean, if this is now, if we're pretty much clear that this is U.S. cyber espionage, if the CIA or the NSA wouldn't have gone to Microsoft and said, you know, the world is using Windows. Now, that's another real discussion point is the fact that, you know, the entire world is using, I mean, dependent upon operating system software from a company based in Seattle, United States.

**Leo:** Right. I mean, these machines, Stuxnet is a Windows virus.

**Steve:** Yes, exclusively.

**Leo:** Yeah.

**Steve:** And here, here there are, you know, Macs are available worldwide. Linux in all of its flavors. It would be, I mean, way safer for all foreign governments not to use a U.S.-based, U.S.-created operating system, yet they all are. There was a blurb I saw in a Kaspersky blog or a summary that said that - oh, no. It was from the Iranian CERT, saying that high-level Iranian officials had been infected by Flame, which is a Windows-only worm. So they're using Windows. And I have to think, wow, okay, that's...

**Leo:** Microsoft would never screw us. Never.

**Steve:** Anyway, so what happened was some clever person, and we'll never know whom, discovered that certificates issued for Microsoft Terminal Server could be used to sign code. And that should have never happened.

**Leo:** So it wasn't Microsoft that signed the code. Somebody who owned a licensed Terminal Server? Is that what I'm hearing?

**Steve:** As I understand it, there was a class of enterprise terminal services, and Microsoft offered a service where you could get certificates from Microsoft and use them to secure Terminal Server. And those certificates, and these are what are now blacklisted, they chained straight up to Microsoft's root authority certificate authority. And there are three of them that are implicated in this. And Microsoft, that's what this emergency out-of-cycle patch is.

**Leo:** So this is a whole - and larger than just Stuxnet. It means any bad guy who had a license for Terminal Services could write certificates.

**Steve:** Yes, yes.

**Leo:** That would be trusted by your browser.

**Steve:** And what no one - yes. And what no - well, no, trusted by Windows.

**Leo:** By Windows.

**Steve:** So these are - because this would - get this, Leo. They also arranged a man-in-the-middle - and this is something we've dreaded forever - a man-in-the-middle attack on Windows Update. And this is one of the propagation mechanisms for Flame in LANs that we discussed last week. There is a component that we actually first talked about as being insecure, you and I discussed it on October 19th of 2006.

**Leo:** We can prove it.

**Steve:** Yeah, it's in the show notes. It's the way I found it. It's Web Proxy Auto Discovery protocol, WPAD. And I talked about how, when you launched IE, there was sometimes a long delay before it started. And I remember watching, I think it might have been Greg, my tech support guy, at GRC's offices at the time. He launched IE, and it sat there stalled for a while. And I said, "Oh, Greg, you've got to turn off auto configure and then IE will launch much faster."

Because what happens is IE sends out a query for WPAD dot and then the machine name dot and then the domain name, looking for a file which contains a script for proxying your communications from inside a corporation. So Flame sets up a server that responds to these queries within a LAN, which then routes the machine's traffic through it, which allows it to get itself in the man-in-the-middle position. And then it maliciously signs its own components and sets these up as Windows Auto Update entities and sends them to those machines as security patches from Microsoft. This is as bad as it gets.

**Leo:** Wow. And this - how long, I mean, could somebody have been using this now?

**Steve:** Yeah. Yeah, I mean, we know - we've discovered it only because - this was heretofore completely unknown. And it's only by reverse-engineering Flame, which has...

**Leo:** Which, by the way, has been around for years. So...

**Steve:** Yes. Yes.

**Leo:** This hole has been around for a long time, I presume.

**Steve:** Well, yeah. The hole's been around forever.

**Leo:** Oh, boy.

**Steve:** What wasn't appreciated was that the Terminal Server certificates would be accepted as code-signing certificates. And that...

**Leo:** So to make this clear, this is not a - Microsoft was not helping with Flame. The authors of Flame discovered...

**Steve:** Well, we don't...

**Leo:** We don't know.

**Steve:** We don't know.

**Leo:** Right.

**Steve:** Because what Microsoft certainly has, plausible deniability. If someone, if a U.S. entity said to Microsoft, we need a way to sign code for national security reasons, and you need plausible deniability - because it will be found. We know it's going to be found. So when it is, we need to have a way that this wasn't you colluding with the U.S. government, which would of course destroy trust in Microsoft forever. So this is, I mean, this is just one more mistake. We've talked about Microsoft mistakes every week. And so, oh, here, oops, sorry about that. And so they immediately revoked those certificates that this code signing depends upon. And that does now shut down the propagation of Flame. Which is no big deal because, get this, Leo, within hours of the discovery of Flame, the entire command-and-control network shut down.

**Leo:** Wow. So they knew it was discovered.

**Steve:** Yes.

**Leo:** That was a response to being discovered.

**Steve:** Yup. Yup.

**Leo:** Oh, wow.

**Steve:** Okay. So that...

**Leo:** So that response, was that a response to news stories?

**Steve:** It was, as I remember, it was Kaspersky's discovery after being asked by the ITU to look into this thing called "Wiper" which was wiping out hard drives. And that still has not been found because of course now they're off pursuing something way more interesting, which is Flame. But, yes, within hours of the announcement of this new thing that was even then still unknown, the 80 domains in a command-and-control network went dark. So...

**Leo:** [Laughing]

**Steve:** It just...

**Leo:** Oh, man. There's a movie here, I'll tell you.

**Steve:** Yes.

**Leo:** I can see the call going out: Shut 'er down.

**Steve:** So I did tweet two links earlier today with the details - I'm going to summarize some of them - from Kaspersky Lab that has been and is continuing to reverse-engineer this. And as I said last week, and here's an example of it, information is going to be coming out incrementally. We'll certainly be covering it because it's fascinating. This is the most sophisticated super cyber weapon espionage tool that has ever been seen. So Kaspersky wrote:

"In collaboration with GoDaddy and OpenDNS, Kaspersky Lab succeeded in sinkholing most of the malicious domains used by Flame's C&C infrastructure. The following details summarize the results of the analysis. First, the Flame C&C infrastructure, which had been operating for years, went offline immediately after Kaspersky Lab disclosed the discovery of the malware's existence last week. Currently there are more than 80 known domains used by Flame for C&C [command-and-control] servers and its related domains, which have been registered between 2008 and 2012. During the last four years, servers hosting the Flame C&C infrastructure moved between multiple locations, including Hong Kong, Turkey, Germany, Poland, Malaysia, Latvia, the United Kingdom, and Switzerland."

**Leo:** Even Switzerland. Huh.

**Steve:** Yeah. "The Flame C&C domains were registered with an impressive list of fake [individuals'] identities and with a variety of registrars, going back as far as 2008. According to Kaspersky Lab's sinkhole, infected users were registered" - so infected users, that is, people who are carrying the Flame virus - "were registered in multiple regions including the Middle East, Europe, North America and Asia Pacific. The Flame attackers seem to have a high interest in PDF, Office, and AutoCAD drawings. The data uploaded to the Flame C&C is encrypted using relatively simple algorithms. Stolen documents are compressed using the open source Zlib and modified PPDM" - they wrote PPDM, but they meant PPMD, which is a partial match statistical compression technology. And they said, "Windows 7 64-bit, which we previously recommended as a good solution against infections with other malware, seems to be effective against Flame." So Flame...

**Leo:** Really. That's interesting.

**Steve:** So the 64-bit version of Windows 7, the malware...

**Leo:** It's that kernel locking?

**Steve:** Well, it's the malware is targeted at 32-bit code.

**Leo:** Ah.

**Steve:** And it's tightly written.

**Leo:** It's nothing Microsoft did.

**Steve:** Right.

**Leo:** Okay. Interesting.

**Steve:** So elsewhere, under "Observations," they wrote, "When a computer is infected with Flame, it uses a default configuration which includes five C&C server domains. Before contacting these servers, the malware validates its Internet connection by trying to access www.microsoft.com, windowsupdate.microsoft.com, and www.verisign.com over HTTPS. If the connection is successful, it will proceed to talk to the C&C domains. Some of the fake identities used to register domains include names such as: Adrien Leroy, Arthur Vangen, George Wirtz..."

**Leo:** Vandelay Industries.

**Steve:** Vandelay.

**Leo:** These are made up, obviously.

**Steve:** Yeah. "Ivan Blix, Jerard Ree, Karel Schmid, Maria Weber, Mark Ploder, Mike Bassett" and so on. "Many of these forged identities have fake addresses in Germany and Austria, notably Vienna. We do not know why," writes Kaspersky, "Vienna was such an attractive choice for the attackers."

**Leo:** Because that's where the sausages come from, of course.

**Steve:** "The fake attackers used addresses of hotels, various shops and organizations, doctors' offices, or simply non-existent addresses." But interestingly, in many cases, the domains were registered to, for example, valid hotel addresses in Germany and Austria. So who knows why? So really interesting stuff we're learning.

**Leo:** This is like spy games. This is good stuff.

**Steve:** And it's true. I mean, it's real. Yikes. So, wow. In other news, this was something that just surfaced last week after we recorded the podcast. But, and I attempted to follow it up, but not assiduously. So I don't have any more details. But a number of people tweeted the news that IE v10 would have Do Not Track enabled by default. Which is huge.

**Leo:** No kidding.

**Steve:** Yeah. And again, as we know, it doesn't proactively prohibit, but it proactively declares that its user does not wish to be tracked. And we're beginning to see maturing behavior on the part of trackers to be responsible in various ways about their behavior relative to the Do Not Track header. So this is just all good news.

Also Apple released an iOS security paper which I've not yet had the chance to go over, but I wanted to let people know that I was aware of it, and I will go over it, see what it says. And if it looks like it's worth a podcast, then we'll give it one. Otherwise I'm sure I'll at least summarize it because it looks like it had lots of interesting stuff. And that touches on another story I'll be talking about in a second where iOS, due to it being the most secure platform available, is pulling the greatest dollar amount in the sale of exploits from hackers who find them to organizations that want them.

**Leo:** Wow.

**Steve:** And government agencies have become the top bidder for these exploits.

**Leo:** Great.

**Steve:** I know. It just gets crazier. In the news this morning was LinkedIn in the doghouse. LinkedIn a couple days ago was caught somewhat controversially sending the calendar, all the calendar details of people's LinkedIn profiles to LinkedIn's servers. LinkedIn defends themselves, saying, well, yes, because we offer the facility to, again, sort of the social networking model, we'll show you the LinkedIn profiles of everyone you're meeting with before you meet with them. And so to do that we need to know what your meetings are going to be. So it's like, okay, fine. Well, in the meantime, six million...

**Leo:** Yeah, but they didn't tell anybody they were going to do that. They just did it.

**Steve:** No, they didn't, yes.

**Leo:** Geez, you'd think companies would learn.

**Steve:** Yeah. Well, here, speaking of learning, 6,458,020 unsalted SHA-1 hashed LinkedIn passwords were recently posted to the Internet.

**Leo:** Oh, see, now, I didn't worry because I saw the SHA-1, that they were hashed.

**Steve:** They were not salted. And they are being decrypted at a high rate.

**Leo:** Oh.

**Steve:** Because it's not, yeah, every LinkedIn person listening to this, you should immediately change your password.

**Leo:** I'm going to be a honeypot. I just want to see what they do.

**Steve:** Okay.

**Leo:** Because what can they do with my LinkedIn account? Who cares?

**Steve:** Well, okay. So consider what they can do if they can log into your account. Are you using that password anywhere else?

**Leo:** No. I checked immediately to see if I had used a unique password, and I had.

**Steve:** Okay, good.

**Leo:** So, I mean, it'd be interesting to see if I get hacked; right? I mean...

**Steve:** Yeah. Yeah.

**Leo:** I don't really use LinkedIn. In fact, I canceled it recently, and it for some reason did not cancel.

**Steve:** There's a needle in a haystack aspect because they do have 6.5 million people to work from. So what's happened...

**Leo:** Ah. So it's only 1 percent.

**Steve:** Yes. It's a small piece of the entire database. And you are one in 6.5 million. What's happened is people have been looking through the list. And many people are finding the hash of their password in the list. And passwords that are dumb, like "Facebook" or "linkedinsucks," for example, are examples from YCombinator. I posted a link to this YCombinator page where there's a really interesting discussion for anyone who wants to pursue this and look into it [news.ycombinator.com/item?id=4073309]. Because it looks like, after the hackers find a match, they put five - they replace the beginning of the hash with five zeroes so that it will no longer match again, to essentially flag it as, okay, we've reversed this hash. Remember, SHA-1 is among - it's not as bad as MD5. That's worse. But SHA-1 is the worst among the two worst hashes LinkedIn could have used without salting it because extremely high-speed hashing hardware

exists. I mean if the NSA in their new Utah facility are doing anything, it's building massive rainbow tables for SHA-1.

**Leo:** So here's my question. There's a site, LeakedIn.org.

**Steve:** Nice.

**Leo:** That says "Provide your password (which we hash with JavaScript, use source to verify) or an SHA-1 hash of your password below, and we'll check to see if it's in the database."

**Steve:** Cool.

**Leo:** So is this safe? I mean, I'm going to give them my password.

**Steve:** Yes. Yes. I would say it's absolutely safe.

**Leo:** I'm going to change it right away anyway, I guess.

**Steve:** Yes. They're going to do it locally. I would say change your password, then...

**Leo:** Then do it.

**Steve:** ...give them your old password and see if it was there. There were some posts that I saw where people had changed their password three weeks ago, just coincidentally, and their old password was in this list. Meaning...

**Leo:** Oh, so it's an older database.

**Steve:** Yes, meaning that it's at least three weeks old because in this one instance it had that person's prior - presumably he didn't use a password like "Facebook" that would have been in there anyway. But they had their old one and not the new one. Of course, it means nothing not to have the new one because, as you said, it's a small portion of the entire LinkedIn database.

**Leo:** And all the hashes begin with four zeroes, or, I mean...

**Steve:** No.

**Leo:** No.

**Steve:** No. What appears to be happening is that...

**Leo:** Oh, look. Look at this. "Your password was leaked but has not yet been cracked." Okay, I guess. Actually it was 10 percent, 6.5 million out of 64 million. So I had a one in 10 chance.

**Steve:** Ah.

**Leo:** Okay.

**Steve:** And yours was there.

**Leo:** I'm in there, they say.

**Steve:** Wow.

**Leo:** They found my hash. There's my hash.

**Steve:** Yeah. So what happens is, after it's been cracked, the crackers replace the first five characters with zeroes, which SHA-1 would have a very low probability of doing.

**Leo:** Ah, I get it. So that's how they know you've been cracked or not.

**Steve:** Yes, it's a simple flag that allows them to quickly do it. So, wow, yours got reversed. And was that one - was it complex? Was it gobbledy-gook? Or was it something that was like dictionary?

**Leo:** No, it was gobbledy-gook. It was a generated pass.

**Steve:** Wow. And, see, that just demonstrates that SHA-1 is that insecure. It is so fast...

**Leo:** Well, wait a minute. They said they hadn't been cracked yet.

**Steve:** Oh, hadn't. Well, no, but many...

**Leo:** Had not been cracked.

**Steve:** Oh, that's a very good point because things like "Facebook" and "linkedinsucks" and so forth...

**Leo:** Those are easy to crack.

**Steve:** Have been cracked, yes.

**Leo:** Mine was a completely random long password that has apparently not been cracked yet.

**Steve:** Which is as strong as you could get it against SHA-1.

**Leo:** Right.

**Steve:** It would take a brute-force...

**Leo:** So explain to me - you've explained this before - why salting is necessary, why SHA-1 isn't - SHA-1 is secure. It's a secure hash.

**Steve:** It's, okay, it's secure. The problem is that it's old, and it's well known. And many organizations like, well, and once upon a time many operating systems were using it to hash passwords without salt.

**Leo:** Right.

**Steve:** So the NSA could build a table where they manually put in every combination of, like, just start at A, B, C, D, E, standard brute-force password cracking, run it through SHA and record the output, and build a dictionary which they then index in the sequence of the output. So that, when they have a hash, they can look that up in this index and immediately see what password generates the hash. They wouldn't know that that was your password, but they would know that that password generates the same hash, which would then allow them to impersonate somebody using a password that generated that hash. So that would allow them, for example, to log in.

So what salting does is it just - it's like it customizes SHA-1. If you did a pseudorandom salt, meaning that for any password the user puts in, before hashing, you append your own gobbledy-gook to it, then that would generate a different SHA-1 hash than if somebody just put "Facebook" into SHA-1 and got Facebook's SHA-1 hash. So if the bad guys knew what the gobbledy-gook was, they could still do forward attacks. But it's much less likely that the bad guys would know what your salt was than just obtaining the database of passwords. Which looks like...

**Leo:** Now, if the salt was stored with the database, that would be bad.

**Steve:** That would be bad. And again, further dumbness. But hopefully somebody who had the smarts to do salting would understand the need to separate the salt from the database. And in this case we know that it was not salted because you can put "Facebook" into SHA-1 and get the same hash as one sitting there in the LinkedIn database.

**Leo:** So that's how this LeakedIn.org site works. I gave it my password. It ran an SHA-1 hash against it...

**Steve:** Right on your local browser.

**Leo:** ...in JavaScript and then presented me with the hash, which then I said, okay, now search for the hash, and it said, yeah, the hash was in the database.

**Steve:** Exactly.

**Leo:** And so if you could do it forward, you could presumably do it backward. LeakedIn.org.

**Steve:** Yeah.

**Leo:** That was fun. I changed my password. I decided not to be a honeypot.

**Steve:** That's good. Because, I mean, if you - that's incremental. Incremental loss of privacy is still a loss. It's not that you can do it backwards, it's that you can do everything forwards. So you keep putting things in...

**Leo:** You keep trying stuff. Got it.

**Steve:** Yes. Try things in a forward direction and see what comes out.

**Leo:** Which is why my truly random password is going to be very difficult because they would have to try stuff. It would only be bad passwords that would be guessed.

**Steve:** Exactly. And so the bad passwords are all being found fast because that's what they're trying first.

**Leo:** So they'll use presumably a dictionary of some kind where they just try common passwords like "abc123" or "adasdf."

**Steve:** Yeah. And apparently they did try "linkedinsucks" because it's one of the ones that has been cracked.

**Leo:** I bet you a lot of people had that password. So if you did as I did, and I've just done again, I used LastPass to randomly generate a password. It seems highly unlikely that a good random password of sufficient length would be guessed.

**Steve:** Very, very unlikely. And also consider this is not a high-value get anyway.

**Leo:** Right, right. There's no credit card in here.

**Steve:** I mean, you were almost - you were almost not caring if someone did get your password. It's like, eh, let's see if I get hacked. Because it's LinkedIn, who cares?

**Leo:** Who cares, exactly.

**Steve:** So, yeah, exactly. So if it were really high-value database, then first of all one would hope that the security would be better. But then you really would want to change your password. And there would be more motivation on the part of the attackers to crack people's hashes and figure out what their passwords were. But in this case - and we're presuming the usernames were stolen, too. They only posted the passwords. The presumption is they have matching usernames. Thus the reason they're going through all this trouble of tracking these things down. So I think next week we will be saying, well, here's the damage that was done because lots of LinkedIn users are going to find their accounts were hacked.

**Leo:** Yeah. Wow. It's kind of a double strike, as you said, because of this calendar stuff. Maybe, I would guess, a few people would take the opportunity to cancel their LinkedIn account at this point.

**Steve:** Well, yes.

**Leo:** That'd be another thing you could do.

**Steve:** And it is definitely a black eye. It is way, we're way beyond the point where there is any excuse for this being the - I'll put "security" in quotes - the "security" architecture for LinkedIn, a substantial, out-in-the-front-of-the-pack, state-of-the-art, web-based system. To be having unsalted SHA passwords, that's nuts. I mean, it's just like - that's a decade ago technology. All the OSes are off of that. Everyone who's doing security, knows what they're doing, are off of that. So this was just written by somebody in the

beginning who didn't think it was going to amount to much, and it did. Briefly.

**Leo:** Actually LinkedIn can tweet on my behalf, so I am glad I did not let somebody use my account.

**Steve:** Ah, yup.

**Leo:** That wouldn't be good.

**Steve:** That'd not be good.

**Leo:** No.

**Steve:** So there was an article a couple weeks ago that I meant to talk about, and it just sort of fell through the cracks, but I saw it again, and I thought, okay, I just need to mention this. And that was - and this is all in this domain that we're in today, talking about state-sponsored cyberwar. And that was the question of Chinese putting backdoors in our chips. There was a rather inflammatory claim made by a company that reverse-engineers chips by popping the lids off of them and looking at them and essentially figuring out what the schematic is of the integrated chip by peeling off the layers of metallization that glue these chips together. And then they've got technology for automating this. And they made the claim that chips being made in China and installed in U.S. networking equipment had backdoors. Now, the good news...

**Leo:** Well, then. Hello there.

**Steve:** Yes. Now...

**Leo:** That's nice.

**Steve:** So here I am laughing that Iran's government agency people are using Windows that's made in Seattle at Microsoft. At the same time, we obviously have an entire infrastructure in the United States of chips from China.

**Leo:** Now, you'd need physical access to the backdoor; right?

**Steve:** Well, okay. So it now looks like this was a false positive, that in fact the interface that this hardware reverse-engineering company found was - it was a known diagnostic portal into the chip.

**Leo:** Oh. Not malicious, in other words.

**Steve:** Exactly. It was part of the original design, not something put in afterwards. But yes, you would - no, you wouldn't need - presumably there was a crypto key that you could use for accessing this remotely. So I'm not wanting to be too quick to laugh here at foreign governments using United States operating system because we're using chips which have all come from a substantial foreign government, and we don't know what has been done to them. I mean, you'd have to open them up. I mean, the problem is finding out. You have to open them up and reverse-engineer every single chip that you're getting, and that's not feasible.

So what it really says is, just as it's crazy for a nation-state hostile to the United States, like Iran, to be using an operating system developed and sourced by an American company, it is every bit as crazy for the U.S. government and the critical infrastructure in the United States to be using networking hardware which comes from anywhere outside of our own borders. So, yow. I mean, these - I'm getting a real sense...

**Leo:** Everything's made in China, by the way, we should say.

**Steve:** These chickens - I know.

**Leo:** Yeah, these are chips in your phone and everything.

**Steve:** Yup, yup.

**Leo:** Although most of the Apple stuff...

**Steve:** It may have a domestic label on it. It may say Cisco or Linksys or D-Link or Netgear. But every component is of Chinese origin.

**Leo:** Amazing.

**Steve:** Yeah. So what is it, the phrase, "the chickens coming home to roost," Leo, I think is the - it's a little scary.

**Leo:** [Clucking]

**Steve:** So Forbes, a couple days ago, Andy Greenberg sort of is their software malware exploits guy. He did a really interesting article, "Shopping for Zero-Days: A Price List for Hackers' Secret Software Exploits." So this is Forbes.com, that's a real magazine. And this is, again, investigative journalism. ZDNet picked up on the story. Their title of their take on this was "U.S. Government Pays $250,000 for iOS Exploit." And their summary said, "Selling exploits to government agencies is becoming a more and more lucrative business. Hackers get paid anywhere between $5,000 and $250,000 for a security vulnerability."

And Leo, if you'll click the link that I've got there, take a look at that chart because what

the article explains is that there's a hierarchy of payment value where the more difficult the exploits are to get and to find and to create, the more valuable they are on this gray market. And many hackers use third-party go-betweens to negotiate on their behalf with foreign governments. Apparently, Chinese hackers pretty exclusively sell only to the Chinese government. But other hackers are selling to various foreign powers. [http://www.zdnet.com/blog/security/us-government-pays-250000-for-ios-exploit/11044]

**Leo:** And it's big bucks.

**Steve:** And so this...

**Leo:** Could be a quarter million dollars for a really juicy one in iOS.

**Steve:** Yes. I mean, and they're - yes. I mean, that's a quarter million dollars for finding something. Now, some of the terms are really interesting. For example, you would like to know that this bad guy is not selling, is not reselling this to many people. So one of the ways this is set up is that only as long as the exploit is not uncovered do payments continue. So essentially, until it goes public, the malware or the exploit discoverer receives periodic payments from the one-time license that they have made to a foreign government. So clearly it's in their interest not to over-disclose it because it would get overused and then discovered and then all payments stop. And so, for example, it turns out that the Russian mafia that has traditionally been a big buyer of these is no longer able to purchase them at the price they were because they tended to immediately use and abuse them, meaning that they had a very short payment life, and so they didn't generate nearly the revenue for the discoverers. But, I mean, listen to what I'm saying. This is crazy. This says there is a mature...

**Leo:** It's capitalism.

**Steve:** [Laughing] Oh. It's a mature...

**Leo:** It's really amazing.

**Steve:** ...functioning marketplace, a global marketplace for unknown defects in highly used operating system platforms which nation states are purchasing in order to launch and in order to build super cyber weapons for espionage. I mean, Leo...

**Leo:** Makes sense to me.

**Steve:** In the past this would have just been science fiction.

**Leo:** Yeah. Wow.

**Steve:** This is mind-boggling.

**Leo:** Wow.

**Steve:** Yeah. And Bruce Schneier weighed in very soberly four days ago. Oh, no, I'm sorry, more than - on April 2. He cites these articles. He says, "This article talks about legitimate companies buying zero-day exploits, including the fact that 'an undisclosed U.S. government contractor recently paid $250,000 for an iOS exploit.'" And then he quotes it, saying, "The price goes up if the hack is exclusive, works on the latest version of the software, and is unknown to the developer of that particular software." Oh, by the way, Leo, I don't know if you noted that Adobe hacks are the least valuable.

**Leo:** Easy to come by.

**Steve:** Uh-huh.

**Leo:** It's directly related to how hard it is to hack, I think; right?

**Steve:** Exactly.

**Leo:** Because iOS is the top. Bonus. Actually, this would be a - this is a valuable chart, yeah, because just above Adobe Reader, on the bottom, in a way, Mac OS X, then Android, then Flash or Java, Microsoft Word, Windows, Firefox or Safari, Chrome or Internet Explorer, then iOS.

**Steve:** Yup, going up in difficulty and platform size.

**Leo:** This has to be a little tempting to those security researchers. I mean...

**Steve:** Well, Leo, think about it. A quarter million dollars.

**Leo:** It's a lot of money.

**Steve:** Yeah.

**Leo:** Please don't die. Please, I beg of you.

**Steve:** You know. And, I mean, it's our tax dollars hard at work, Leo.

**Leo:** Yeah. Well...

**Steve:** You and I are paying for it because it's a government contractor that's buying these in order to equip our country's cyber weapons. I just - this just makes my eyes cross. Anyway, just to finish this paragraph, "Also, more popular software results in a higher payout. Sometimes the money is paid in installments, which keep coming as long as the hack does not get patched by the original software developer."

And so Bruce continues, "Yes, I know that vendors will pay bounties for exploits. And I'm sure there are a lot of government agencies around the world who want zero-day exploits for both espionage and cyber weapons. But I just don't see that much value in buying an exploit from random hackers around the world." And there he has a point. Except that the economics, as I explained earlier, really do inure to the benefit of a hacker behaving himself. So I can see that, too.

He says, "These things only have value until they're patched, and a known exploit - even if it is just known by the seller - is much more likely to get patched. I can much more easily see a criminal organization deciding that the exploit has significant value before that happens. Government agencies are playing a much longer game. And I would expect that most governments have their own hackers who are finding their own exploits. One, cheaper. And two, only known within that government." So really, really interesting stuff.

**Leo:** Really is. Surprising.

**Steve:** And I've got email from Kevin Rose.

**Leo:** Now, that can't be THE Kevin Rose.

**Steve:** And it wasn't. I asked. He said...

**Leo:** I know Kevin uses SpinRite, but I just - I don't - yeah.

**Steve:** Yeah, he said, "A very fast SpinRite recovery," sent it on May 14th. And he wrote, this Kevin Rose wrote, "While SpinRite has a history of going slow, it can also fix issues rather quickly. On an older computer I had BSOD at boot-up, unmountable boot volume. The first thing I did was boot it into my trusty copy of SpinRite and ran it on Level 2. 40 minutes later it was complete, with one unrecovered sector, though the report did say that most of the data had been recovered. I was able to boot back into Windows and back up all the data from the 40GB hard drive, then run SpinRite again on Level 4. The computer is now working normally as my VPN server." So when I read that, I thought, huh, well, guy's got a VPN server. That kind of sounds maybe like Kevin.

**Leo:** Yeah.

**Steve:** So I said - I wrote back, I said, "Hey, Kevin. I don't know whether this is THE

Kevin Rose, but either way, thanks for sharing your success story. I'm so very glad that SpinRite was able to help you." And he replied, "Nope, this is not the more famous Kevin Rose. I wonder if he," he says, "I wonder if he uses SpinRite."

**Leo:** Oh, he does. I know he does.

**Steve:** And you've confirmed that.

**Leo:** Well, he used it on The Screensavers. We used it all the time.

**Steve:** Ah, right. And so he said, "This is one of the best" - this is the non-Kevin Rose Kevin Rose said, "This is one of the best programs I have ever used. While it is not used as often as many other programs, when it is used, it is easily worth five times its price. To date, SpinRite has saved a total of five hard drives for me: two desktop hard drives, 40GB and a terabyte drive; one 320GB hard drive in my old laptop; and two external hard drives which, oddly enough, were the two drives I have that did not have a fan in the external enclosure. Using eSATA to the drive shows that during normal use they were within 1 degree C of their overheating temperature." So anyway, non-Kevin, thank you very much for your SpinRite story. And thank you, Leo, for confirming that THE Kevin Rose is also a SpinRite user.

**Leo:** Of course he is.

**Steve:** That's cool.

**Leo:** All right, Steve. I've got questions.

**Steve:** Cool.

**Leo:** 12 of them, starting with Scott Maser in Colorado Springs with a question about the "DynaStat" screen on SpinRite: Steve, I'm running SpinRite on a drive that failed miserably. It was a Western Digital Network Storage drive that had some data I am trying to recover. Backups. The SpinRite - see, this is - this probably was a backup drive. It was his network storage drive. And yet...

**Steve:** They go bad, too.

**Leo:** They go bad, too. One copy of anything, whatever drive it's on, is still not a backup. The SpinRite screen is covered with a lot of "B's" - that means bad - and the time remaining counter keeps going up as it hits the "B" sectors. I fully expect to let this run for a while just to see what I get. And by the way, Scott, could be a while. Could be a long time. I've been trying to decipher the DynaStat screen. I'm confused with the bit numbering. I see bits 0 through 32. Isn't that one too many bits if I'm

looking at things using a 32-bit word? Is there a reason there are 33 bits on the screen, Steve? 33 bits?

**Steve:** Okay. There is a reason. That's how many would fit.

**Leo:** Oh [laughing].

**Steve:** Okay. The DynaStat screen, it's mostly just eyewash. I mean, it is showing you what's going on. But mostly it's - "DynaStat" stands for Dynamic Statistics, and it's a technique that is unique to SpinRite, which allows SpinRite to often recover unreadable data where at no time is the sector readable, but SpinRite can figure out what it was when it was readable. And because this takes some time, it can take up to 2,000 samples of the sector, I had to have SpinRite show you something while it was running and doing this. And so the DynaStat...

**Leo:** Are you saying it's eye candy?

**Steve:** Well, yeah. I mean, it's true. It's true eye candy. But you shouldn't try to, like, figure out what it means.

**Leo:** Right.

**Steve:** It's the, first of all, a sector that's 512 bytes is 4096 bits in a stream. Bits are stored, not as, like, bytes at a time, like 8 bits abreast. They are actually stored in a linear bit string that has no byte boundaries. So that's why 33 bits is as good as 34 or 31 or 32. It's just SpinRite is showing you where in the 4096-bit stream, the problem begins that SpinRite has located. And it's able actually to zero in on the problem and start working on it. And so what looks sort of like an oscilloscope diagram are the statistical probabilities of the bits from the first bad bit being zeros and ones. So it's, I mean, what it's showing you is true. It's sort of a - it's a viewport into the database, the statistical database that SpinRite is building over time as it analyzes that sector in order to try to recover its data.

So that's a little tutorial on the DynaStat screen. It's, again, it's mostly something for you to look at. Just like, be patient, SpinRite's going to recover this sector if there's any way possible. And in fact it's able to even - SpinRite will give you the data out of those 4096 bits that it can, even if it can't get all of them, which is, again, another very unique thing about SpinRite that often allows recovery to occur even when the sector could never, ever be read correctly and corrected. For example, if this was a chunk of a directory, well, you might get most of the files that were linked from that branch of the directory, and that's more valuable than getting none of them. So, or like having the directory stop at that part of the file system. So that's just a reason why SpinRite's able to so often pull off the miracles that it is.

**Leo:** Question 2 from Jason Varner, Pennsylvania, USA. Jason says, "I wanted to

mention AES Crypt awesomeness. Awesome. Dear Steve, after hearing you discuss Duplicati on a recent episode of Security Now!, I decided to try it out. It's another one of those cloud storage systems; right?

**Steve:** It's a frontend for S3 that is absolutely multiplatform, which is really nice.

**Leo:** While I wasn't particularly impressed by the Linux (Ubuntu in my case) version of the GUI interface, looking into Duplicati did lead to the discovery of the awesome AES Crypt piece of software [aescrypt.com]. As a relatively recent Security Now! listener, I don't know if you've ever discussed AES Crypt before, but I wanted to make sure you were aware of this elegant, simple solution for AES 256-bit file encryption. I'm assuming this encryption package provided is solid - Linux users ONLY have the option of downloading and compiling the source code - but wanted to ask for your feedback on AES Crypt. As my need for a remote backup of my data is not so sensitive that daily or even weekly backups are necessary, I am now employing the following completely free and rather simple remote backup process (hopefully TNO compliant):

1. Make a ZIP file of directories to be remotely backed up, with the date of the snapshot included in the filename, e.g., BACKUP.20120529.zip.

2. Encrypt that ZIP file using AES Crypt and a strong password, which gives you the same file with a .aes extension.

3. Upload that file to the BACKUPS folder on Google Drive.

While this solution isn't the most accommodating to the need for frequent backups, i.e., the entire backup file has to be uploaded each time, and you have to think to do it and blah blah blah, it fits my needs. I guess he could write a cron job to do this.

**Steve:** Yeah.

**Leo:** Your feedback on the AES Crypt software and my process would be greatly appreciated. Jason Varner.

**Steve:** So I wanted to make sure we pointed people to AES Crypt. I use it.

**Leo:** Oh.

**Steve:** And I like it.

**Leo:** Better than TrueCrypt?

**Steve:** Well, it's entirely different. It is a very simple, lightweight, bulletproof, AES cipher application, and cross platform: Windows, Mac, it's available in Java, in C-sharp, also for

Linux. It's open source. So what it is, I mean, we've talked a lot about what AES encryption is. This is simply a utility to give end users access to AES 256-bit file encryption. So it's just a - it's as simple as you use this in the same way that you use ZIP to zip up a bunch of files, you use this to encrypt a file. It asks you for a password. And that password is hashed and then used as the key for the encryption. And no force on Earth, as far as we know, if you use a strong password, is able to decrypt it. So it's absolutely bulletproof. Under Windows, the app does a whole bunch of nice things with - it's got a nice UI. And it also will put things in the context menu so you can right-click on a file and say "AES Crypt This," and it will encrypt it and decrypt it and so forth.

So it's such a great - I wanted to bring it up, thank Jason for mentioning it. This came up in the context of Duplicati because our listeners will remember, if they looked into Duplicati further, that Duplicati bundles the file format into their backend, that is, the files that Duplicati uses to store at Amazon is AES Crypt compatibly encrypted because the other thing that AES Crypt has done is to publish their file format. So the Duplicati people said, hey, rather than reinventing the wheel, let's use the AES Crypt file format and the cipher, which is as good as anything else.

And the cool advantage to that is, if anything, for any reason at all, you ever couldn't use Duplicati, you still have full access to those files because you could use the standalone AES Crypt to, after you bring them back from Amazon, to decrypt them. So, and it's also just a really nice standalone encryption tool. So I just - I thought I wanted to give everyone a pointer to that.

**Leo:** Ranmadhu in Australia wonders: How does Google know you're using DNSChanger? That's the malware that Google is now detecting and announcing. In Security Now! 354 you were talking about how Google has come up with a method to determine if someone's using the wrong DNS servers. I'm completely at a loss as to how they can do this. I wasn't aware that a remote server could tell which DNS server a client was using. It would be great if you could elaborate. Thanks, and keep up the great work with the show.

**Steve:** Well, so I have not looked specifically to see how Google does this. But let's remember that Google is, if anyone in the world is, is running script on your browser. I mean, Google's whole - the whole focus is turning your browser into a desktop surrogate, essentially. So I was thinking, from my remembering JavaScript, and it's been a while since I've coded anything in JavaScript, I don't think you can get low level enough for JavaScript to see the IP address of a DNS entry that is looked up. But, for example, Google could include some items on the page which are a domain that is resolved by the bad guys.

Now, if your DNS server, whatever DNS server you were using, resolved one of the bad IPs for one of those domains, then that would tell Google you were using DNSChanger DNS servers still. But I don't know that it's possible for JavaScript to determine what the IP is. Now, maybe they're playing some games beyond JavaScript. Or all that would be necessary would be if Google found something that those servers returned differently, because of their maliciousness, than good servers. And I don't know whether there is, like, something Google knows that we don't know.

But so my point is that, if Google's page asked for a resource on a domain which was like a different size or in any way different than what a valid non-DNSChanger, DNS server would return, then they could certainly detect - they could certainly tell from the nature of what was looked up by that DNS server, whether it was DNSChanger or not. So

essentially, when you're running - even though the Google server can't tell, you need to remember that when we're going to Google, we're running Google's script on our browser. And then there's all kinds of things that they're able to do.

**Leo:** It's all sorts of magic.

**Steve:** Yeah.

**Leo:** So it would require that JavaScript would return this information. Returns a lot of information.

**Steve:** It does. I don't think, I mean, Java, the Java language definitely could do this. I don't think...

**Leo:** JavaScript is what they use.

**Steve:** ...a JavaScript allows you to look up the IP for a domain name. But maybe it does, in which case it'd be even simpler to just see if it's among those, the IPs. They might send it back to Google, and Google looks to see if it's among those. But again, just something, any behavior that was different about those servers, and specifically like what they returned, that would be enough to tip off that you were using those malicious servers.

**Leo:** Really interesting, isn't it.

**Steve:** Yeah.

**Leo:** Very interesting.

**Steve:** Love the technology, Leo, love the tech.

**Leo:** Yeah, yeah, yeah. Quib. Our next question is from Quib in Southern California, who says: Greetings from the past! Steve, I discovered your podcast a few weeks ago, absolutely love it. Decided to take a casual sip from the fire hose of the episode archive. Well, it is a fire hose. There are, what, 356 total episodes.

**Steve:** Yeah.

**Leo:** He says: I'm currently on 74, which is back in the Vista pre-release days. Wow.

**Steve:** It is a bit of a Wayback Machine that he has.

**Leo:** Yeah, I forgot we go back that far. You sound so optimistic about how the new architecture will protect the OS from all sorts of nasty things. I've only listened to a few new episodes, but I do know you're still doggedly hanging onto XP. I never would have guessed that five years from where I am right now, you would be so against the new platform. For the benefit of those who haven't yet listened to the other episodes, the remaining episodes, would you give the listeners a brief overview of what went wrong? Did Microsoft get lax and start letting every passerby drop code into the kernel? Did the creators of malware find a way to bust through the protection? Or was it really an improvement for security, but other irritating issues kept you from making the switch? Thanks for the show. You are a great service to the Intertubes, says Quib.

**Steve:** So, okay. Microsoft clearly improved security dramatically from XP to Vista, and fixed the things that they really didn't do that well, sort of maybe went overboard with Vista, in 7, making 7 more friendly. Yet we don't see attacks which are only effective against XP. All the attacks that we see are always effective against all of them. So when you think about it, there isn't a differentiation. I'm not seeing anything that gets 7 that doesn't also get XP. Why? Because it's still the same operating system. Microsoft comes up with new layers of eye candy and new UI features, but nothing fundamentally changes. I mean, yes, Address Space Layout Randomization gets better, and DEP is more strongly enforced, and a few things like that. But they can't really change much without breaking all of the legacy stuff. So they're limited in what they're able to do. And you could argue that they're sort of running out of things to do at this point. So first of all, looking back at all the patches we've discussed in the last year, nothing is XP only. I can't think of anything that only affected XP.

**Leo:** How interesting.

**Steve:** It's always all of them. And so here I am using XP, not seeing any effective improvement. I mean, these are, oh, look, we added a bunch of security features. Everybody move. It's like, okay, let's see. Nothing showed up.

**Leo:** Nothing.

**Steve:** Nothing seems to be actually more secure. And I don't see anything that I want over on Windows, lord knows on Vista, but even on 7. I mean, it looks different, but it's just in my way more. So it isn't demonstrating better security. Now, that will change in two years or three years, whenever it is that patches stop being offered for SP3. So at that point I'll think, okay. Either the bad guys will have moved off to Windows 8, and no one will even be bothering to attack XP anymore because it'll be more like Windows 98 is, for which none of these things are effective because it'll just have enough different DNA that it can't be infected. Or maybe I'll switch. I'm not sure. But at the moment, XP is the same as 7 in every way I can tell. Everything I want to do is compatible with XP still. So there's no incompatibility problems. And there's no demonstrated actual effective increase in security. So why would I move?

**Leo:** Let me...

**Steve:** By the way, these are all free for me. I'm an MSDN subscriber. I pay Microsoft...

**Leo:** So you could move. It's not like you're spending more money.

**Steve:** ...$2,700 a year to have access to all their OSes. So, yeah, it's free. Costs me nothing to move. But except my time because I'm fighting with 7. I mean, it just looks like a toy to me more and more. So, I mean, I felt that way about XP compared to Windows 2000. So I'm just a curmudgeon by nature. But at this point I'm just digging my heels in. It's like, unh-unh. This thing works.

**Leo:** Well, at some point you're going to have to. I think there's only two years left in the update cycle.

**Steve:** Only. Leo, come on.

**Leo:** Hey, this is from five years ago this guy is writing, so two years is nothing. It's like the blink of an eye.

**Steve:** Yup.

**Leo:** Bob Harris in New England mentions that the TrueCrypt/Dropbox trick could corrupt users' data: Steve, in Security Now! Episode 350 - we've got a bunch of oldsters here - you mentioned a TrueCrypt/Dropbox trick which could maybe result in some bad things. The problem is if the folder sync utility manages to transfer an actively mounted container file system, then it would be possible for the user to mount the container file system on a different machine or machines concurrently with the original. The danger here is there is no coordination between the systems. The algorithms used to allocate new files and storage are going to be the same, when the systems are the same OS. So if two or more systems try to use the mounted container file, creating new files or allocating storage, they are likely to choose the same blocks. The last system to sync their changes wins - maybe.

It's more likely that the files will be lost, or only partially there, or have the mixed contents of several files in them. It also is possible that the file system metadata could become corrupt such that even a chkdsk or fsck cannot repair the file system, losing all the users' data. And even if the user promises they will never mount the container file system on more than one system concurrently, hey, accidents happen. It might only take one "oops!" to corrupt the data. This dangerous trick could be tried with any number of container file systems, e.g., an encrypted Mac OS X Sparse Bundle, some of which might just allow Dropbox or similar folder synching utilities to transfer data for an active mounted file system. I'm not sure I understand this, Steve. Maybe you'd better explain.

**Steve:** Yeah. So this was so obvious to me that it made me shudder. And it was definitely worth mentioning to people, although I think we're protected from it. First of all, so what he's talking about is that - I know that some people are doing this. But I believe that you cannot both have the file in Dropbox and mounted by TrueCrypt. So

what he's talking about is the danger of taking a file and making it a container file for TrueCrypt, which gives you, when mounted, a drive letter, and having it also sitting in Dropbox and visible to other machines. Now - or copied to other machines, as I understand that's what Dropbox does is it clones it to your drives on other machines, where it's accessible. It is a nasty hack, the idea of mounting a drive which is a file that's in a shared resource that was never really designed to be shared. But Dropbox must have provision for handling desynchronized changes among files. So this might be handled by TrueCrypt. But, I mean, sorry, by Dropbox. Or it may just not be possible to have TrueCrypt mount it when the file is, like, opened by somebody else, so that there's some sort of exclusivity.

**Leo:** But Bob, there is some file locking. I think so.

**Steve:** There must be file locking. But Bob is correct. Imagine the horror of this file, which is nothing but the sectors of a file system, being simultaneously available to different machines that each believe it is their file system. That is, there's no notion that this is simultaneously - any other machine has access to its sectors. Because that's what TrueCrypt is doing. Its mounting the file system means that the blocks of data in the file are virtual sectors of a virtual hard disk.

So the reason I was made to shudder is that, I mean, it's like a classic, like cache coherency problem or cache access conflict. Any file system is excruciatingly careful to only allow access through a given port so that all of the activities behind the scenes are synchronized through that one viewpoint. And this would be like having multiple views into the raw data. And if that were possible, nothing would prevent multiple operating systems from just going in and, like, allocating the same sectors for new files and just overwriting each other when they wrote those back. Which would be horrific.

Now, the fact that this apparently isn't a problem and doesn't happen makes me think, as you said, Leo, there's - and I think for Dropbox to be effective they have to have somehow managed concurrent access to shared files. So something resolves this for people. But I just wanted to make sure, for anyone who is relying on this, that you maybe test this or make sure that you're safe in using this, what is otherwise kind of a cool hack.

**Leo:** What Dropbox does for me, if there's a conflict, is it creates a new file that says - let's see if I can find one here, a conflict file.

**Steve:** Ah, so it branches off.

**Leo:** It branches. And it says there's a conflict.

**Steve:** Okay.

**Leo:** Which isn't that helpful because, you know.

**Steve:** Yeah, then you...

**Leo:** How do you merge the two?

**Steve:** Essentially you've forked your file system.

**Leo:** Yeah, and that means you're forked.

**Steve:** Yeah.

**Leo:** Yes. Moving along. Creighton in Arizona points us to a new CAPTCHA solution. Steve, the following site may interest you. A company is unveiling a series of drag-and-drop logic puzzles to prove you're human. Now didn't I just read that CAPTCHA...

**Steve:** Yes.

**Leo:** ...had been broken?

**Steve:** Yes. Google has been having a real problem with their...

**Leo:** reCAPTCHA.

**Steve:** Yeah, their reCAPTCHA solution.

**Leo:** Which they got from Carnegie Mellon and I really like.

**Steve:** Yeah. And it was great. But, you know. And we've discussed CAPTCHA a lot because the whole problem of bots getting increasingly clever is prevalent. We talked about there was that one that was kind of like a waving flag one that I liked a lot. This one is, again, another take. It's AreYouAHuman.com, just demonstrates their technique. I'm not that impressed with it. I mean, what impressed me, Leo, is that it's so hard to tell.

**Leo:** I don't know, I've got a stack of pancakes.

**Steve:** Okay.

**Leo:** I've got some tools, including a daisy, a saw, butter, and maple syrup. And I guess the presumption is, if you figure out the butter and the maple syrup work better than the saw and the daisy...

**Steve:** Yeah, I got a pizza when I tried it yesterday. And I got pepperoni and cheese, I think, maybe tomato sauce, and a few non things. And they're all kind of drifting around.

**Leo:** That's kind of cool.

**Steve:** I mean, it is. But again, I wonder, okay, how hard is that?

**Leo:** Well, here's the way you break this stuff. You use a human.

**Steve:** Yeah, that's a problem.

**Leo:** You put up a porn site, or a fake porn site.

**Steve:** You redirect to - yes, right.

**Leo:** And you put this CAPTCHA on it as an iFrame.

**Steve:** Or you pay people in Russia...

**Leo:** Or you just pay people.

**Steve:** There is a site where you can make money, it's a Russian site, you sit there and you solve these CAPTCHAs in real time, and you're only paid if you solve it quickly and if it's correct. Then you get some money added to your account. It's a little micropayment system. And so it's - and these are people, this is quote, "a job," unquote. And that's what they're doing. And these are the human front-end for a bot network which is then using this to create accounts under - in fact it was Gmail you read about because Google Mail is generally not - has a lot of spam. And so it's generally regarded as safe. And unfortunately, they've been having problems with their CAPTCHAs. You're right. Ultimately there's no solution.

**Leo:** CAPTCHA is stupid. Stop using it. Thank you.

**Steve:** And it's annoying. Because, I mean, I'm often, I look, and I go what the heck is that? I don't know what...

**Leo:** If it's a high-value target, it's easy to break. And if it's not, stop bugging me.

**Steve:** Well, isn't this an interesting problem that we're having, that it isn't - well, but think about it. It's also interesting that it is actually difficult to differentiate a human from a computer.

**Leo:** Right. It's called the Turing Test. In reverse.

**Steve:** Yes, they could do so much.

**Leo:** DarthNader in Minneapolis says Password Haystacks are too good of an idea: Steve, remember when you were talking about passwords? Well, it seems your idea was too good because, like, many good things, it's been foiled by those who could most benefit from it. My national bank chain made me change my password today, and their rules now include one about how you cannot use the same character three or more times in a row, eliminating my ability to use a string of the same character as a haystack. They also told me that I cannot change my password more than once in a 24-hour period, eliminating my ability to change my password until my desired password was out of the "recent passwords" list. Well, that's good. Well, I could still do it, but it would take over a week to do. So I guess he's one of those guys who wants to reuse his old password?

**Steve:** Yeah.

**Leo:** Next time you have a good idea, please don't take it to the mainstream media because, once the public knows about it, so does my bank. I'd like to be able to use your good ideas. Actually, well, all right, I'll let you answer this one.

**Steve:** Well, I was going to say, first of all, I doubt, I mean, it's flattering, but I doubt that...

**Leo:** They're just trying to keep you from doing 1111111111.

**Steve:** Yes. GRC and Password Haystacks probably isn't the reason the bank made this change. And I would note that no one said you had to use dot dot dot dot dot dot dot dot...

**Leo:** Right.

**Steve:** ...to pad.

**Leo:** I don't use repeating characters.

**Steve:** Exactly. And I don't either. That was just an example of a way of padding. So you can definitely pad with something a little more clever. And I'm not going to offer any suggestions because anyone listening can figure out their own scheme, and I'd rather not put one out there that people go, oh, I'll use that one. So, yeah. Certainly there are ways around that. And it sounds to me like the bank has got good security policies. These are the sort of things we want them to do.

**Leo:** Somebody in the chatroom just sent me a password policy from - I guess it's from the state of Texas. Actually it looks like Texas State. Or maybe, yeah, it's portal.computerscience.oag.state.texas.us. Password has to be exactly eight characters long. I'm not sure what that's all about.

**Steve:** Oh, my god.

**Leo:** It must contain at least one letter, one number, and one special character. But the only special characters allowed are @, #, and $. A special character cannot be located in the first or last position. Okay, that means two through seven. Two of the same characters sitting next to each other are considered to be a set. No sets are allowed. Avoid using names such as your name, userID, or the name of your company or employer. Other words that cannot be used are Texas, child, and the months of the year. A new password cannot be too similar to the previous password. A password can be changed voluntarily once in a 15-day period. The previous eight passwords cannot be reused. This is just brain-dead.

**Steve:** Oh, my god.

**Leo:** Some good stuff, but mostly stupid.

**Steve:** Okay. So let's wrap up with this next question, which is the one I referred to earlier.

**Leo:** Okay.

**Steve:** Which is a good place for us to stop.

**Leo:** Our last question. Oh, you don't want to do the motorcycle question? Which one did you refer to earlier? Do you mean the question...

**Steve:** #8.

**Leo:** #8, okay. We'll get to the rest another time.

**Steve:** Yes.

**Leo:** Mike in Thailand. And this one is regarding Flamer, Skywiper, and Infrastructure Systems Security: Steve, thank you and Leo for the very informative shows. In the past you performed a very detailed analysis of Stuxnet, which I found more useful than many Industrial Control Systems analysis. I work with ICS systems

and see that much of the IT in use and thinking is five to 10 years behind the times. I have found it very difficult and frustrating to get people to really understand the risks. Working outside the U.S., I see things from a more global, interconnected perspective. Australia sees all this as the start of cyber war. Mikko Hypponen, chief research officer at F-Secure, sees a future of cyber race.

My question is: What is your thought on the big-picture direction of all this? That's what we, your listeners, want to know. What do you think it means for the future? Thanks, and best of luck to you and Leo. Are we in a cyber war? So this gets to the question that we were talking about, whether...

**Steve:** Yes.

**Leo:** ...President Obama did the right thing to aggressively pursue Olympic Games.

**Steve:** Yeah. I would say to escalate what may have been going on. And, I mean, it is, it's a real question, I think. We know that, from stories that we've covered, that there appear to be incursions which have never been admitted by entities of some cloth in China who are poking at and probing and often breaking into our U.S. infrastructure. The Chinese government always disavows any responsibility or affiliation and so forth, although these things generally seem to be coming from China, which has a lot of people and a lot of Internet connections. And so statistically, maybe, even if they were random, that would be the case. But it's a really good question.

My problem is that - I used the expression earlier, this notion of the chickens coming home to roost. By that I mean our technology is incredibly porous. Our security is really bad. I mean, we launch platforms that are written quickly, that are generally late, we're behind schedule. Management says is it secure? The programmers say, well, yeah, we think so. We'd like to have a few more weeks. And they say no, no, ship it now, we'll fix it later. I mean, there's that kind of approach to commercial entities that have the wrong motivations for publishing software which is too important, arguably, to be wrong. I mean, when we hear that the drone control system is using Windows and got infected by a thumb drive...

**Leo:** Oh, dear.

**Steve:** ...you think, okay, wait a minute.

**Leo:** That can't be good.

**Steve:** You know? And, I mean, I worry that Iran may be working to purify uranium for the purpose of building a bomb. They say they only want it for domestic power production.

**Leo:** Yeah, yeah, sure.

**Steve:** And so they're running centrifuges which we apparently are able to screw up. Once again, remember, these were not even on the network, but it didn't matter because Stuxnet could jump from thumb drive to machine and back in order to infect the control systems. Well, we're using these SCADA control systems for our dams and our nuclear reactors and huge mission-critical systems. And they're connected to the Internet because, oh, it's convenient to be able to log in. Anyway, so...

**Leo:** Although these were air-gapped. That's why they had to use thumb drives.

**Steve:** Yup. And it didn't matter. It got across the air gap.

**Leo:** Didn't matter. They were still using Windows.

**Steve:** Yup, exactly. They were using Windows. So, and, I mean, for a long time our listeners would send me pictures of the ATM machine with the dialogue box...

**Leo:** The Blue Screen of Death, yeah.

**Steve:** Either a BSOD or, more often, a notice popped up with a button you had to click, but there was no mouse to click with. And famously, the big, huge Vegas kiosks will have, like, a Windows message that is...

**Leo:** Right, error, error, error.

**Steve:** ...come up on top of - oh, god, yeah. So I don't know. I mean, this is during the course of this podcast, which, what, we are now in our seventh year, or seven and a half year? Are we on our eighth year? I don't remember. Anyway, during the course of this podcast...

**Leo:** Way too many, yes.

**Steve:** ...this has gone from theoretical to way past real. We are now in real with Flamer and before that with Stuxnet. And, I mean, I remember - remember when Stuxnet first began to happen, I was initially skeptical. It's like, eh, we need to wait for more data. I don't want to jump to any conclusions here. Well, now we know. And there is no question that Flamer is beyond - the capabilities in Flamer are beyond an individual or a small group. This is, when you've got repurposed valid Microsoft security certificates, and someone figured out or arranged that they could be used to code sign in order to allow Windows Update to be intercepted, I mean, how many times have we worried...

**Leo:** That's wild, isn't it? That is just wild.

**Steve:** Yeah, how many times have we worried that Windows Update might be

vulnerable, and all of our Windows machines would be downloading malicious code? Well, Flamer does that.

**Leo:** I love it that they can use Windows Update to update themselves. I mean, I don't love it, but I just think that's pretty amazing.

**Steve:** You, your system has an obsolete version of the malware.

**Leo:** Of the malware. Would you like to update? Well, so, and the debate we had was really whether the feds - whether it was right for the U.S. to pursue this cyberwarfare strategy.

**Steve:** An undeclared aggression.

**Leo:** And I think that, given the alternative - and this started in the Bush White House, where Vice President Cheney was urging bombing attacks on Iran, which would have really been destabilizing in the region, and I think they decided, well, instead of bombing these plants, let's try this.

**Steve:** We'll do something that won't hurt lots of people.

**Leo:** Right.

**Steve:** It'll be more, a little bit more like a drone attack, if I might use a...

**Leo:** There's some question to its efficacy. I mean, they did get people - apparently it was efficient to the point that the Iranian scientists took all of the centrifuges offline because they couldn't figure out why they were failing. But I don't know if it really slowed down the enrichment process in any significant way. Certainly not as much as a bomb might have.

**Steve:** Yeah, estimates, optimistic estimates in that case were maybe it knocked them back 18 months at the most.

**Leo:** Right. So it's not a huge...

**Steve:** But it certainly didn't shut down the program.

**Leo:** Right. So, and we have said time and time again, oh, terrible, we shouldn't do it, nobody should do cyberwarfare, those darn Chinese are doing it, well, now we know everybody's doing it.

**Steve:** Yeah.

**Leo:** And I guess my opinion is it's kind of the way it is. Should we - it's not like poison gas, which we all agree, all civilized governments agree not to use. It's...

**Steve:** And landmines are not...

**Leo:** Or bio, bio weapons.

**Steve:** Yeah.

**Leo:** It isn't like that, although I guess if you use a cyber attack on significant important infrastructure like the electrical grid, and you brought it down, it would have some deleterious effects. I just think that this is the way war is - war is not a good thing, but you can't bury your head in the sand.

**Steve:** And we do have, for example, we have CIA agents that are operating covertly which are sort of the same sort of thing.

**Leo:** It's just how it is.

**Steve:** Embassies, foreign embassies are known to be basically satellite spy centers...

**Leo:** We've been doing that for ages.

**Steve:** ...of governments.

**Leo:** Right.

**Steve:** I think maybe it's - what's a little unnerving, Leo, is it's moved into our territory. I mean, it's moved into the purview of this podcast.

**Leo:** Well, yeah.

**Steve:** Where it's become real.

**Leo:** Secure yourself. But I think that cyberwarfare is inevitable, and I think that it would be foolish of the U.S. government to ignore it and not to participate out of some moral high ground. I just don't.

**Steve:** I don't think we need to worry about that.

**Leo:** I think that that's a weapon we need. It's a weapon we need.

**Steve:** We don't need to be - worry about any moral high ground.

**Leo:** [Laughing] There's plenty of worse stuff. And I think it's an appropriate weapon. I do.

**Steve:** Yeah.

**Leo:** So I guess, after chewing on it, I don't think it's inappropriate to use this weapon. In fact, in some ways, not bad.

**Steve:** And it's not - this isn't...

**Leo:** It's nonlethal in some cases.

**Steve:** Flamer was just espionage. As far as we know, from what's known at this point, it looks like it was an information-gathering tool. It was taking screen shots and capturing keystrokes and looking for AutoCAD DXF files. And, now, what we don't know, and this could easily change our opinion or amplify our opinion, we don't know how much incredibly valuable intelligence it was gathering. Somewhere there are probably really unhappy people who were involved in turning off the command-and-control network four hours after Kaspersky announced their discovery because something vital to presumably Western intelligence gathering was taken offline. It went dark. They lost what may well have been a fantastic source of intelligence. So we're looking at it sort of from a, oh, what does it do and how does it work. We know nothing about, in detail, what it actually gathered. And in four years, boy, it may have just been a phenomenal success.

**Leo:** Steve Gibson is at GRC.com. That's his home. That's where SpinRite lives, the world's best hard drive maintenance utility. And of course a lot of freebies he gives away because he's just a nice guy. As well as 16Kb versions of this show in audio and full transcriptions. If you want the video or the - what is that you're showing there? What is that? What is that? Is that next week?

**Steve:** No, that's the...

**Leo:** What are you up to?

**Steve:** That's the little prototype for the ketone breathalyzer.

**Leo:** [Laughing] You madman. You've done it. Does it work?

**Steve:** It's on its way.

**Leo:** He's breadboarding a ketone analyzer. Well, it's about time.

**Steve:** Yeah, exactly.

**Leo:** Wow.

**Steve:** Because I don't have enough things on...

**Leo:** What chip do you use to detect the presence of ketones? Is there a sensor?

**Steve:** There are volatile gas sensors which will detect ethanol and also acetone. The problem is that they all - they're very sensitive to temperature and humidity, and our breath is both hot and moist. So the signal I'm looking for is minuscule compared to the noise, which is temperature and humidity. So I have a second sensor which is exactly the same technology, but designed to detect methane instead. And so the idea is that the common mode response will be humidity and temperature, and the differential response will be the content of gases that differ between the two sensors. So anyway, I'm just at the beginning of...

**Leo:** What a fun challenge.

**Steve:** ...of experimenting. It may be that I cannot find - it may be that breath is just too hostile because of its temperature and humidity. But I'm going to - I'm working to very quickly determine, one way or the other, because I am just so tired of - my hands are just raw from poking them in order to take blood several times a day, which I have been doing.

**Leo:** Several times a day?

**Steve:** Oh, yeah, yeah, because I'm spending serious money on these $5 ketone blood tests in order to monitor my ketones and get a sense for where they are. I would - I can't wait to be able to, you know, to blow into something. And if it works, we'll, I mean, I'm not going to go into production. People don't have to worry about me disappearing...

**Leo:** I think you should open source this. You should give this away, yeah.

**Steve:** I'm absolutely going to - I'm going to open design it, and we will probably do a,

what's the site where these things are crowd-sourced?

**Leo:** Something -dables, expendables...

**Steve:** No, I'm thinking, shoot, I'm drawing a blank. I've said it to many people. It's the - it's where everybody says, hey, I have an interest in this, and you put up a pledge against - Kickstarter.

**Leo:** Oh, Kickstarter.

**Steve:** That's what I'm trying to say.

**Leo:** Oh, you could Kickstart it.

**Steve:** So the idea would be, all of our listeners, given it's possible, it would be a battery-operated, handheld thing.

**Leo:** I think you'd do quite well.

**Steve:** I call it the Ketoflute, since it would use audio.

**Leo:** Ketoflute.

**Steve:** The Ketoflute.

**Leo:** [Whistling]

**Steve:** And so, I mean, for anyone who's doing this - and I've got to say, Leo, it's good to put this at the end of the podcast, so anybody who doesn't care can hit stop. They're done.

**Leo:** They're already tuned out, yeah.

**Steve:** I'm getting so much feedback from our listeners who we have helped with these Over the Sugar Hill podcasts. Several people have lost 35 pounds. Their blood tests have normalized. One guy from Scotland said, "I smell funny, thanks to you, Steve."

**Leo:** Me, too.

**Steve:** But he just loves what his body is doing.

**Leo:** Yeah, it's awesome.

**Steve:** So it was really a good thing.

**Leo:** Yeah. And you feel like the keto strips are not as accurate as you'd like them to be? Is that the issue?

**Steve:** Well, they don't continue working. There's an adaptation in your muscles that begin to burn the acetoacetate.

**Leo:** Oh, that's what's happened.

**Steve:** Yes. So you're still in...

**Leo:** Okay. That explains it.

**Steve:** Yes, you're still in ketosis, but the strips no longer register. There are expendable, but unfortunately very expensive, they're $5 per test, they're like the glucose tests.

**Leo:** That's why you do the blood tests, yeah, yeah.

**Steve:** That's why, yeah, those are the little deals. And then I do a weekly urinalysis, and it freaks out that I've got ketones in my urine. It's like, yes, I know, I can smell it.

**Leo:** [Laughing] The Sugar Hill, two specials that we did, if you want to know more on the TWiT Specials feed with Steve Gibson about the ketogenic diet. And he has recommendations for reading there. But you can also go to GRC.com/health for links. And if you want audio of a higher quality, or video, we've got that at TWiT.tv/sn. We do this show every Wednesday, 11:00 a.m. Pacific, 2:00 p.m. Eastern time, on TWiT. Please watch live. We'd love having you as a live audience. But if you miss it, don't worry, you have plenty of ways to watch after the fact. Do subscribe. I think that's a great thing to do.

**Steve:** And don't forget that we do get questions from our listeners. Every two weeks we go through a sampling of them, and those go to GRC.com/feedback. So that's how to get stuff to me. And of course I do keep an eye on my Twitter feed where I get a lot of great stuff, feedback from our listeners, too.

**Leo:** @SG...

**Steve:** @SGgrc.

**Leo:** ...grc. Thank you, Steve.

**Steve:** Thanks, Leo.

**Leo:** See you next week on Security Now!.